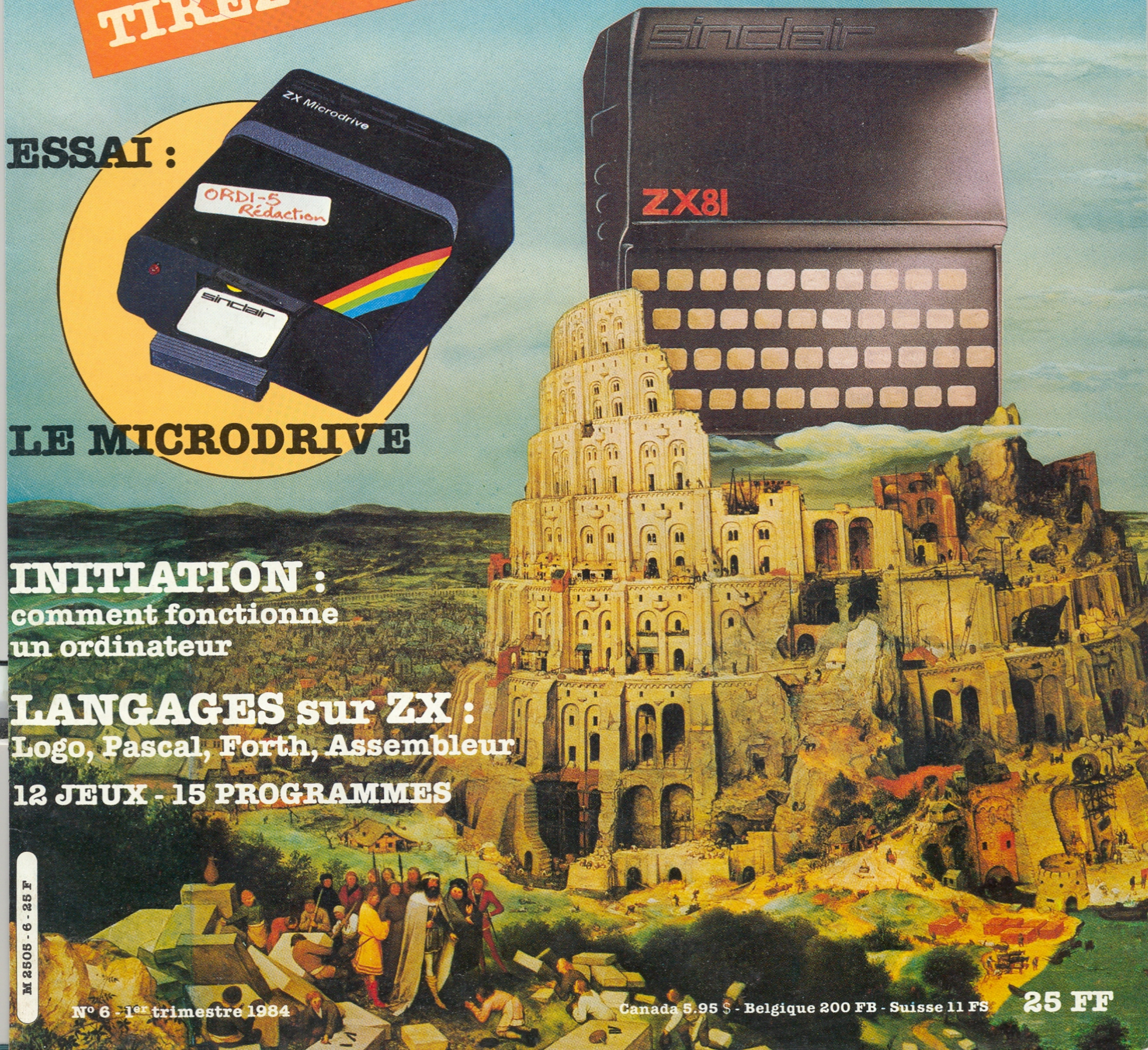
ISSN 0755-4249

N° 6

ORDI-5

TIREZ PLUS DE VOTRE SINCLAIR



ESSAI :

LE MICRODRIVE

INITIATION :
comment fonctionne un ordinateur

LANGAGES sur ZX :
Logo, Pascal, Forth, Assembleur

12 JEUX - 15 PROGRAMMES

M 2505 - 6 - 25 F

N° 6 - 1er trimestre 1984

Canada 5.95 $ - Belgique 200 FB - Suisse 11 FS

25 FF

DOSSIER HORS SÉRIE

L'ORDINATEUR INDIVIDUEL

Ce dossier présente une sélection des meilleurs programmes publiés dans l'Ordinateur individuel

50 PROGRAMMES BASIC

Jeux, enseignement, professionnels, utilitaires, etc. et un tableau comparatif de tous les Basic pour pouvoir utiliser les programmes sur : Alice, Apple, Atari 400, Atom, BBC, Commodore 64, DAI, Dragon 32, Hector, Lynx, MZ 80 A, Oric, PET/CBM, TO 7, TI 99, TRS 80, Vic 20, ZX 81, ZX Spectrum.

en vente dans tous les kiosques
à partir du 10 décembre

Canada : 3,95 $C ; Belgique : 285 FB ; Suisse 11 FS

35 F

ORDI-5

N° 6

Editorial



La Tour de Babel est une expression qui commence à n'être plus vraie pour les ordinateurs familiaux du moins ceux que nous connaissons bien. Le ZX Spectrum et le ZX81 acceptent désormais grâce à des logiciels performants (il est vrai assez consommateurs de mémoire) de comprendre les programmes en Pascal, en Forth et même en Logo au-delà du BASIC. Sans parler bien sûr de l'assembleur qui n'est que la façon polie de s'adresser au microprocesseur que l'on peut interpeller directement par le langage machine.

Avec l'étude d'une cassette (Pascal), le démontage d'un jeu expliqué (assembleur et Forth), un simulateur de tortue (Logo), avec toutes ces choses ORDI-5 a choisi de vous initier aux secrets de ces différents langages par le chemin des écoliers (en récréation).

Encore de l'initiation : comment marche un ZX, quels sont les éléments essentiels d'un ordinateur, cela aussi figure en bonne place dans la revue avec bien sûr plein de jeux, plein de programmes, et un compte rendu complet sur le microdrive tant attendu.

Joyeux Noël et à bientôt.

PS : ORDI-5 aimerait bien voir ses petits souliers remplis de programmes 1K spirituels et bien tournés que vous aurez composés pour lui.

ORDI-5

Sommaire

Rédacteur en chef : Alain Pinaud.
Editeur : Jean-Pierre Nizard.
Directeur de la publication : Bernard Savonet.
Conseiller technique : Xavier Linant de Bellefonds.
Maquette : Sylvine Dautref.
Secrétariat : Nicole Alleman.
Illustration : Christian Augé, Yvan Boussion, Dominique Delpierre, Stéphane Druais, Isabelle Mounier. *Couverture :* Sylvie Taugourdeau sur une idée de Christian Augé.
Ont collaboré à ce numéro : Kaarina Alain, Tristan D'Amico, Freddie Blin, Jean-Claude Bouman, Jean Casenave, Bruno Célerier, B. Clergeot, F. Delaquaize, F. Falque, Patrick Guculle, Didier Gumy, J.-C. Heur, Marcel Henrot, B. Lacoste, Morandini, Michel Murbach, Jean-Luc Papillon, G. Pedroli, Pesce, Claude Remy, Sherer, J.-M. Schimpft, Régis Senegou.
Rédaction et abonnements : Editrace, 8, rue Saint-Marc, 75002 Paris.
Régie publicitaire : Force 7, Anne Jourdan, 5, place du Colonel-Fabien, 75491 Paris Cedex 10
Tél. : (1) 240 22 01
Diffusion auprès des boutiques informatiques et des libraires : PSI Diffusion, B.P. 86, 77402 Lagny/Marne Cedex.

ORDI-5

LE MAGAZINE DES UTILISATEURS DE SINCLAIR



Si vous utilisez un ordinateur SINCLAIR (ZX 81*, ZX 80* ou Spectrum*) ou si vous comptez en acheter un, sachez que la revue **ORDI-5** a été créée pour vous. Indépendant de tout constructeur ou importateur, **ORDI-5** vous fournit quatre fois par an des programmes, des conseils, des astuces, de nouvelles idées d'utilisation. **ORDI-5** teste pour vous en toute objectivité et indépendance les produits matériels et logiciels adaptables sur votre SINCLAIR. **ORDI-5** vous tient au courant de toutes les nouveautés susceptibles de vous intéresser.

ORDI-5 n'est pas en vente chez les marchands de journaux. Pour vous abonner ou pour recevoir 1 numéro, il vous suffit de nous retourner le bon de commande ci-dessous.

Abonnez-vous, vous économiserez 20 %

ORDI-5, le complément indispensable de votre ZX

*marques déposées

BON DE COMMANDE

à retourner à ORDI-5, 8 rue Saint-Marc 75002 PARIS

Nom ______________________ Profession ______________________

Adresse ____

Pays ______________ Code postal ______________ Ville ______________

☐ Je désire recevoir le n° 1 ☐ le n° 2 ☐ le n° 3 ☐ le n° 4 ☐ le n° 5 ☐ le n° 6 ☐ de ORDI-5 (prix d'un n° 25 FF ; Etranger** 30 FF ; par avion 40 FF).

☐ Je désire m'abonner à ORDI-5 pour 1 an, 4 n^{os}, à partir du n° 1 ☐, du n° 2 ☐, du n° 3 ☐, du n° 4 ☐, du n° 5 ☐, du n° 6 ☐ (tarif France 80 FF ; Etranger** 90 FF ; par avion 160 FF). (Actuellement ORDI-5 est trimestriel).

Ci-joint mon règlement indispensable par chèque bancaire ☐ chèque postal ☐ virement ☐

** Pour les pays autres que la France, utiliser un virement en FF compte Crédit Lyonnais Paris n° 30002 00402 8455 J. Les frais de virement sont à la charge de l'acheteur



Le petit livre du Spectrum
par Trevor Toms
Destiné aux utilisateurs du ZX Spectrum, ce livre présente douze applications ludiques exploitant au maximum les possibilités de graphismes, de sons et de couleurs du dernier-né de chez Sinclair. Les lecteurs apprécieront tout particulièrement un chapitre clair et complet sur l'utilisation des codes machines, ainsi que celui concernant l'emploi de la ROM pour ses propres programmes.
168 pages - 82,00 FF/635,00 FB

Boîte à outils pour ZX 81, Spectrum et Timex 1000 et 2000 NOUVEAU
par Marcel Henrot
Cet ouvrage n'a pas la prétention de livrer le secret des grandes applications informatiques, mais bien de mettre à la disposition des utilisateurs de petits prorammes tout prêts qui leur permettront de résoudre de nombreux problèmes de la vie quotidienne.
128 pages - 35,00 F/250,00 FB

102 programmes pour Sinclair ZX et Timex
Sinclair ZX 81 et ZX Spectrum-Timex 1000, 1500 et 2000
par Jacques Deconchat
Apprendre en se distrayant, tel est l'objectif de ce livre. Au fil de ces 102 programmes de jeux il vous guidera dans l'exploration du Basic Sinclair. Les programmes sont classés par niveaux, chaque niveau faisant appel à de nouvelles connaissances. Des instructions correspondant à chaque niveau sont présentées et commentées avec des remarques concernant les points spécifiques du ZX81 et du ZX Spectrum. Tous les jeux sont décrits, et les programmes abondamment commentés ; un exemple d'exécution est fourni pour chaque version (ZX81 et ZX Spectrum). Ecrits pour le ZX81 dans sa version de base (1 K octet), pour le ZX Spectrum et pour les Timex 1000, 1500 et 2000.
240 pages - 102,00 FF/785,00 FB

Pratique du ZX Spectrum - *Tome 2*
par Marcel Henrot
Cet ouvrage, consacré au langage machine, est accessible à tous ceux qui ont assimilé la programmation Basic approfondie. La corrélation entre les commandes Basic et les codes machine a été la préoccupation principale de l'auteur, qui amène progressivement le lecteur à la compréhension de routines de plus en plus complexes. Les trois aspects principaux du Spectrum : le son, la couleur et la haute résolution font l'objet de nombreuses routines dans tous les chapitres. Le lecteur attentif élaborera des programmes personnels en code machine quand il aura refermé ce livre.
168 pages - 82,00 FF /635,00 FB

La pratique du ZX Spectrum
par Xavier Linant de Bellefonds
Tome 1
Cet ouvrage exploite toutes les possibilités du ZX Spectrum dans le domaine de la programmation avancée. L'auteur y expose d'une manière soigneusement graduée : la définition de caractères et les fonctions de chaîne, la haute définition graphique et la puissance de calcul (graphiques en trois dimensions...), la couleur et le son, les techniques d'accès direct à la mémoire et l'utilisation du langage machine.
168 pages - 82,00 FF/635,00 FB

Clefs pour le ZX Spectrum NOUVEAU
par Jean-François Sehan
Comment gagner du temps ?...
Sans en perdre à glaner de-ci de-là tous les renseignements techniques dont vous avez besoin pour bien utiliser votre Sinclair.
Les "Clefs" c'est : • La liste des instructions Basic commentées • Les mnémoniques de l'assembleur Z80 et leurs codes objets • Les points d'entrée de la ROM Basic • Des explications sur les variables système • Une liste d'astuces pour mieux utiliser l'écran, les cassettes et les programmes en langage machine.
112 pages - 82,00 F/635,00 FB

P.S.I.

P.S.I. DIFFUSION
BP 86 - 77402 Lagny-S/Marne Cedex
FRANCE
Téléphone (6) 006.44.35

P.S.I. BENELUX
5, avenue de la Ferme Rose
1180 Bruxelles
BELGIQUE
Téléphone (2) 345.08.50

P.S.I. SUISSE
Case postale
Route neuve 1
1701 Fribourg
SUISSE
Tél. : (037) 23.18.28

au Canada
SCE Inc.
65, avenue Hillside
Montréal (Westmount)
Québec H3Z1W1
Tél. : (514) 935.13.14

Prix valables jusqu'au 31.12.83

Envoyer ce bon accompagné de votre règlement à P.S.I. DIFFUSION ou, pour la Belgique et le Luxembourg à P.S.I. BENELUX ou, pour la Suisse à P.S.I. SUISSE.

ORS 12

DESIGNATION	NOMBRE	PRIX
par avion : ajouter 8 FF (75 FB) par livre	TOTAL	

Signature (obligatoire pour paiement par carte de crédit)

☐ Paiement par chèque joint ☐ Paiement en FF par carte bleue VISA (à P.S.I. DIFFUSION uniquement) paiement supérieur à 50 FF

N° ______________ Date d'expiration ____

NOM ______________ PRENOM ______________

rue ______________ n° ______

Code postal ______ Ville ______________

AGRAPH

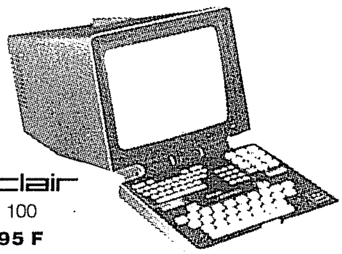
ORLÉANS

45000

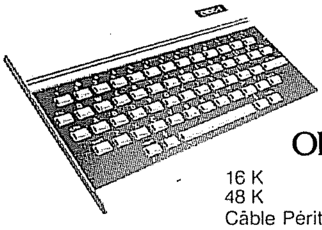
98, rue du Fg Saint-Jean

Tél. (38) **62.05.17**

Magasin OUVERT
tous les jours
de 9 h à 12 h - de 14 h à 19 h
sauf Dimanche et Lundi matin

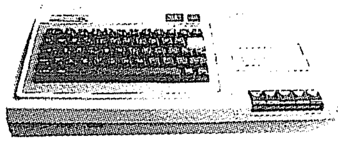


sinclair
VP 100
2995 F



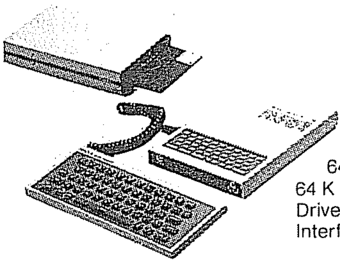
ORIC-1

16 K	**1670 F**
48 K	**2140 F**
Câble Péritel	**110 F**
Alimentation Péritel	**70 F**



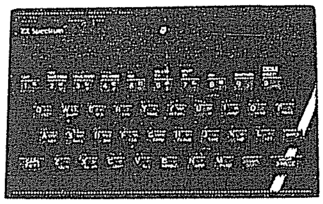
hector

16 K	**2450 F**
48 K HR	**4390 F**
48 K HRX	**4950 F**
Adaptation Basic III résident	**950 F**



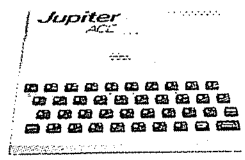
MULTITECH

64 K Pal.	**2995 F**
64 K Péritel Sécam	**3390 F**
Drive	**2985 F**
Interface drive	**435 F**

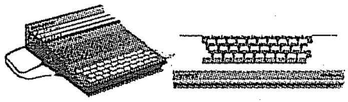


sinclair SPECTRUM

16 K Péritel	**1850 F**
48 K Péritel	**2325 F**
16 K Pal.	**1480 F**
48 K Pal.	**1965 F**

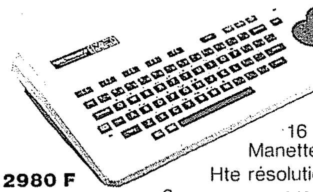


Jupiter ACE **940 F**

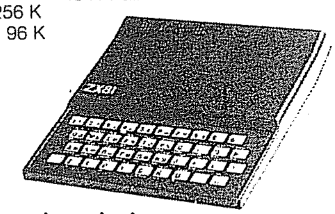


Extension MEMOTECH

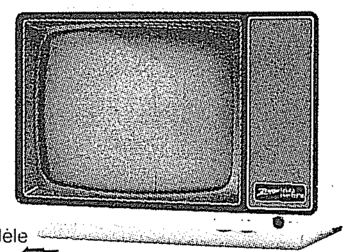
16 K Ram	**380 F**
32 K Ram	**545 F**
64 K Ram	**795 F**
HRG	**495 F**
IF Centronics	**445 F**
Câble IF	**170 F**
Clavier	**545 F**
Mémotext	**445 F**
Mémocalc	**445 F**
Rom Assembleur	**445 F**



SPECTRAVIDEO
SV 318
32 K Ram extensible à 256 K
32 K Rom extensible à 96 K
Basic SV microsoft
Acces direct CP/M
(standard 80 col.)
16 couleurs
Manette de jeux intégrée
Hte résolution (256 × 192 points)
3 canaux sonores, 8 octaves
2980 F

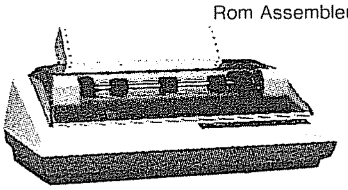


sinclair ZX 81 **580 F**



Ancien modèle
ZVM 121 **ZENITH**

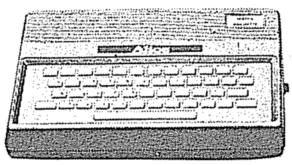
Nouveau modèle
ZVM 123 E Moniteur vert 12 pouces **1250 F**
ZVM 122 E Ambre 12 pouces **1375 F**



SEIKOSHA

GP 250 X
(64 caractères programmables)
interface parallèle centronisc
RS 2 32 C DIN
2600 F

GP 100 MARK II
2450 F



MATRA ET HACHETTE

4 K Ram 9 couleurs, Péritel, son, sortie K7 et imprimante, complet avec son alimentation
1190 F

Nous sommes en mesure de fournir toutes les extensions, logiciels, librairie, concernant tous ces produits.

Documentation contre 2 timbres à 2 F

BON DE COMMANDE

Offre valable jusqu'au 31/12/83
à retourner à E.S.C.

Nom .... Prénom ....
Adresse complète ....
Code postal .... Ville ....
Tél .... Signature
Date ....

Quantité	Désignation	Prix unitaire	Prix Total
		TOTAL	

Mode de règlement
☐ Chèque bancaire joint
☐ Chèque postal joint
☐ Mandat-lettre joint

Participation frais port et emballage + 40 F

PubliVal

courrier des lecteurs



Depuis quelque temps, il semble bien que la curiosité de nos lecteurs évolue nettement vers la constitution interne du ZX81, ainsi que vers le raccordement d'accessoires. Indépendamment des articles paraissant dans nos colonnes sur ces sujets passionnants (surtout compte tenu de l'auréole de mystère dont les entoure Sinclair), le meilleur ouvrage que nous ayons eu l'occasion de lire sur la question est sans conteste :
"THE INS AND OUTS OF THE TIME TS 1000 AND ZX81", écrit par Don Thomasson et publié par Melbourne House.
Il est hélas fort difficle de se procurer en France ce petit livre dont nous avions fait l'analyse dans ORDI-5 n° 4.
Voici les coordonnées de son éditeur, qui ne refuse guère les commandes émanant de l'étranger :

MELBOURNE HOUSE Publishers Ltd.
Glebe Cottage Glebe House
Station Road Cheddington
LEIGHTON BUSSARD
BEDFORDSHIRE LU7 7NA (Grande Bretagne)

Un mandat-poste international est la solution la plus commode pour règler le montant qui sera réclamé avant expédition de l'ouvrage.

UN CIRCUIT INTEGRE PROPRE A SINCLAIR

Notre lecteur Pascal Aillerie trouverait par exemple dans ce livre l'explication de son échec lorsqu'il a tenté de faire accéder la machine à un nouveau générateur de caractères implanté par lui en RAM.
Nous avons vu, en effet dans notre article du n° 5, que la modification du contenu de registre I n'autorise un changement de jeu de caractères que sur l'imprimante, à condition encore d'ajouter un petit montage additionnel.
La raison de tout cela tient à ce que le générateur de caractères de la ROM est adressé, lors de la création d'une image TV, non par le microprocesseur ZX80, mais par le fameux "circuit intégré Sinclair", tandis que le ZX80 lui, travaille en liaison avec la RAM !
C'est d'ailleurs pour cela que neuf résistances séparent la ROM de la RAM sur les lignes AØ à A8 du bus d'adresses. Monsieur Vincent Lacour trouvera encore bien d'autres informations dans ce livre, sur les autres fonctions du circuit Sinclair, qui l'intrigue plus particulièrement. Ce gros circuit intégré appartient à la famille des "réseaux prédiffusés" ou, en anglais, "uncommited logic arrays", souvent abrégé en ULA.
Il s'agit de réseaux de portes logiques dont on peut modifier l'interconnexion au moyen de procédés industriels simples, tout comme on programme une ROM.
Il est donc tout à fait vain de chercher les caractéristiques de ce circuit dans un quelconque recueil de produits standards, puisqu'il n'est destiné qu'au seul ZX81.
Seules des recherches patientes, confirmées par des recoupements avec le schéma du ZX80 (ancêtre du ZX81 réalisé à partir de boîtiers TTL courants), peuvent conduire à une compréhension satisfaisante du fonctionnement de ce circuit chargé de bien des tâches :
- génération de la fréquence d'horloge générale, au moyen d'un filtre céramique de récepteur TV (6,5 MHz).
- division par 2 pour obtenir l'horloge à 3,25 MHz du microprocesseur Z80 A.
- gestion des entrées-sorties cassette
- gestion du clavier
- sélections mémoire
- génération des lignes de l'image TV, en liaison avec le générateur de caractères de la ROM.

A PROPOS DE L'IMPRIMANTE

Beaucoup de questions également sur l'imprimante Sinclair : comme tous les produits de la marque, microdrive compris sans oublier le téléviseur ultra-plat, ce périphérique fait appel à des techniques très anticonformistes.
Une sorte de courroie intérieure supporte deux petites pointes élastiques défilant à grande vitesse, mais en parfait synchronisme avec l'avance du papier, puisque les deux mouvements sont dus à un même moteur lié à un train d'engrenages nylon.
Comme ces pointes sont diamétralement opposées, l'une attaque un bord du papier juste au moment où la précédente quitte l'autre.
On obtient ainsi un balayage permanent du papier métallisé, fort similaire à celui d'un image TV.
C'est d'ailleurs pour cela que cette imprimante arrive à recopier fidèlement l'écran TV, même en présence de graphismes haute ou basse résolution, ce qui est sans précédent compte tenu du prix pratiqué !
Comme la mécanique est bien loin de suivre l'électronique côté vitesse, imprimante et ordinateur échangent en permanence des signaux de synchronisation par le port 251 ou FB. Un avantage peu connu de cette petite imprimante est son incomparable fiabilité en matière de transfert d'information : comme les données d'impression lui parviennent point par point, chaque caractère imprimé nécessite huit octets, répartis sur huit lignes successives.

Si l'un de ces octets se trouve perdu ou altéré, il n'en résultera qu'une légère déformation du caractère, alors que sur une imprimante recevant directement les codes ASCII (un seul octet par caractère), le caractère serait irrémédiablement perdu.
Le principal inconvénient de ce petit périphérique est la nature du papier employé, absolument inutilisable pour la correspondance ou de la facturation.
Il existe, du moins en Grande Bretagne, une foule d'interfaces permettant de raccorder des imprimantes papier ordinaire au ZX81.
Le prix de ces accessoires varie beaucoup selon que le logiciel d'accompagnement est logé dans une ROM incorporée, ou tout simplement fourni sur cassette.
Tout un transcodage est en effet indispensable entre les codes Sinclair, et les octets ASCII que réclame, par exemple, une imprimante conforme au standard CENTRONICS.

En France, nous ne disposons guère que de l'interfaçage MEMOTECH, et d'un adaptateur spécialement étudié par TEKELEC pour son imprimante "best seller", la GP 100 SEIKOSHA.
Il s'agit du nec plus ultra, permettant au ZX81 de traiter les minuscules et même certains caractères graphiques, mais cette interface coûte... deux fois le prix du ZX81 et l'imprimante près de cinq fois ! Autant dire qu'il est plus sage de changer d'ordinateur, ce qui se ressentira par la même occasion au niveau du clavier, bien peu adapté à des applications "bureautiques".

LA PERITELEVISION

Plusieurs questions de vidéo également, émanant de lecteurs souhaitant connecter leur ZX81 à un magnétophone, ou un téléviseur muni d'une prise "péritélévision" non standard à un providentiel ZX Spectrum.
Dans le premier cas, rien n'est plus simple puisque le ZX81 peut être considéré exactement comme une antenne TV délivrant une émission sur le canal 36 (environ). Pour une adaptation directement en vidéo, on se reportera plutôt à cette rubrique parue dans notre numéro 2 : ce genre de problème avait, à l'époque, été traité dans le détail.
Pour ce qui est des prises "péritels" non conformes à la norme "SCART" en vigueur actuellement, signalons que des fabricants réputés de connecteurs proposent aux revendeurs TV toute une gamme de cordons précâblés capables de résoudre la plupart des problèmes, et même d'adapter, sans la moindre intervention interne, un prise normalisée à tout récepteur n'en comportant pas (coût 500 F environ).
Tout bricolage dans ce domaine demande une grande prudence, car les variantes sont innombrables pour les prises non normalisées.
L'assitance d'une agent de la marque du récepteur est donc vivement souhaitable.
A titre de rappel, nous reproduisons cependant ci-après le brochage des prises péritélévision normalisées :

Patrick Gueulle

→ Dans le dernier numéro d'ORDI-5 vous avez publié un dessin intitulé MARTHE. Pourriez-vous m'envoyer le listing de ce programme ? (Joël Ganzin).

* Derrière l'image de Marthe, il n'y a pas un programme particulier mais une table de points PLOTtés. Pour PLOTter Marthe, sans jeu de mot vulgaire, son créateur l'a d'abord dessinée sur du papier millimètré et a ensuite introduit une table dans l'ordinateur, chacun des points ayant une coordonnée horizontale et une coordonnée verticale - il existe des calques millimètrés qui vous permettront de transcrire ainsi n'importe quel motif graphique qui vous paraîtra beau.

→ Quel dommage que le Spectrum n'ait pas une télécommande de magnétophone ! il doit bien y avoir un moyen de décoder les ordres de chargement et de fin de programme (Bernard Breton).

* Si le Spectrum n'est pas fourni avec un jack de télécommande de magnétophone c'est que, dès sa conception on a imaginé que la meilleure mémoire de masse pour lui ne serait pas une cassette mais une microdisquette, laquelle s'arrête d'elle-même dès la fin de tout chargement.

→ Comment distinguer tout de suite si vos programmes sont faits pour le ZX81 ou pour le Spectrum ? Vous ne dites rien en tête d'article (Yves Dusonchet).

* La différence est immédiate : lorsqu'un programme comporte plusieurs instructions par ligne ou des minuscules, c'est un programme Spectrum.

→ Je suis au regret de ne pouvoir renouveler mon abonnement à ORDI-5, ayant renoncé à acquérir un Spectrum devant le peu d'enthousiasme déployé par son importateur (G. Lepesant).

→

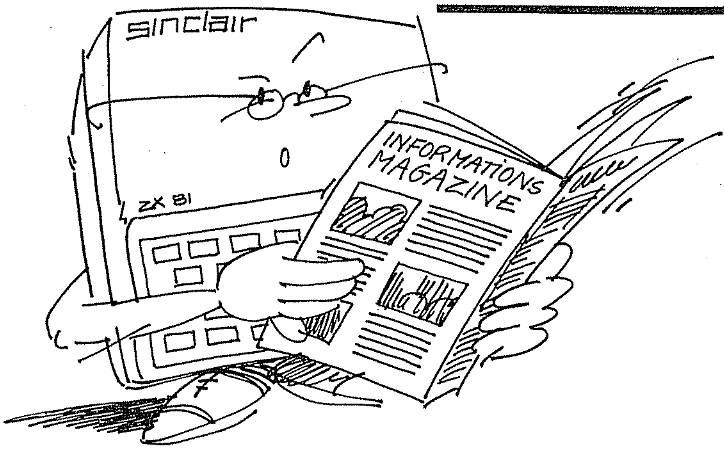
* Cher Monsieur, nous citons votre lettre car elle est caractéristique d'un très abondant courrier : l'extrême lenteur du Spectrum à pénétrer en France est plus le fait de la fabrication que de l'importateur. Le grand succès de cet ordinateur dans les pays Anglo-Saxons a conduit Sinclair à considérer le marché français comme résiduel dans les premiers temps. Cet état de choses devrait changer dans le tournant 83-84 pour notre soulagement à tous.

→ J'aimerais savoir si l'on peut vendre ou acheter des cassettes du commerce à d'autres personnes - les vraies cassettes, bien sûr ? (Gérard Demoustier).

* Certainement, seule la duplication de cassette est interdite car elle signifie reproduction et que la propriété littéraire et artistique qui protège les programmes s'oppose à la reproduction. Quand vous achetez une cassette de programme ce n'est pas le programme que vous achetez mais le droit à l'usage de ce programme : vous pouvez parfaitement revendre ce droit à usage à quelqu'un d'autre. Mais la plupart du temps n'est-ce-pas, avant de cèder la cassette originale n'est-on pas tenté de la dupliquer pour ses propres besoins ?...
Si vous voulez en savoir plus sur la protection juridique des programmes, vous pouvez consulter "L'Informatique et le Droit" de Xavier Linant de Bellefonds Que sais-je, n° 1923 (c'est un ami !).

→ Pourquoi ne parlez-vous pas davantage du Cyborg que la publicité présente comme la disquette adaptée au Sinclair ?

* Nous attendons pour vous parler du système Cyborg que le système soit effectivement disponible, ce qui à notre connaissance n'est toujours pas le cas malgré les promesses faites à ceux qui ont payé d'avance. Nous ne pouvons en effet vous parler que des matériels que nous avons jugés sur pièces, ce qui nous l'espérons sera sans doute le cas bientôt.



Magazine

QUID NOVI SPECTRUM ?

Un 16 bits Sinclair ? On sait de bonne source que la firme est en train de préparer pour l'horizon 86 un ordinateur 16 bits. Le ZX80, le ZX81, le ZX Spectrum étaient tous basés sur un micro processeur à 8 bits, c'est-à-dire capable d'adresser seulement 64K de mémoire. Avec 16 bits cette limitation n'existe plus et le gain de rapidité est sensible. Déjà on peut considérer que les ordinateurs de table qui n'ont que 8 bits sont quelque peu périmés.

Un ZX83 ? Il en est question mais les informations sont assez contradictoires à ce sujet. Il pourrait s'agir d'un micro ordinateur portable conçu autour du microprocesseur 8051 d'Intel et d'un écran plat.

La batterie des extensions maison du Spectrum s'est récemment enrichie avec une deuxième interface, l'interface dite "2" qui permet de transformer l'ordinateur en système de jeu particulièrement souple par l'enfichage de cartouches ROM préenregistrées. Cette formule, qui a été imposée par le marché américain, procède d'une toute autre philosophie que l'interface "1" que nous analysons dans ce numéro. Conformément à la "loi Sinclair" les cartouches préenregistrées sont les plus petites du genre : elles ressemblent pour la taille à des extensions MEM telles qu'on les enfiche sous les ordinateurs de poche japonais. Le seul problème est que chaque cartouche, ayant fait l'objet d'une étude poussée, coûte évidemment cher à l'achat, environ 200 F, ce qui est un prix ordinaire comparé à d'autres constructeurs mais un prix élevé pour les produits Sinclair. L'interface n° 2 supporte évidemment les manettes de jeu.

Vous aimeriez faire lire directement par votre Spectrum les programmes que vous avez écrits pour un ZX81 sans avoir à tout retranscrire à partir du clavier avec les modifications nécessaires ? Une cassette de Amersham Software limited vous y autorise : il suffit de charger le Spectrum avec cette cassette pour qu'il lise automatiquement les signaux en provenance d'une cassette ZX81. Cette cassette n'est pas coûteuse : 4 livres seulement en Angleterre.

Memotech qui a fait des extensions nombreuses pour toute la série ZX81 a décidé de se lancer dans la fabrication des ordinateurs et propose un MTX 500 ; un appareil certainement très complet mais combien plus cher que le Spectrum !

MASH va commercialiser pour la rentrée deux nouveaux produits pour ZX81 : une mémoire RAM 16K (Econotech) 345 F TTC (cette mémoire a été étudiée pour éviter toute rupture de contact) et une mémoire RAM 64K ultra plate aux connecteurs professionnels plaqué or. Ces produits sont diffusés par la société INNELEC.

INNELEC distribue également en France les produits Basicare que nous avons déjà décrits dans ces colonnes et qui se présente, d'une manière assez professionnelle sous forme d'empilements (gamme organic micro) une dizaine et plus d'extensions peuvent ainsi se superposer avec des mémoires permanentes, des EPROM, des unités entrées sorties, etc.

Un nouveau distributeur pour les produits Sinclair : Silicone Valley à Nantes et à Angers.

CA VA BIEN MERCI !

Le 13 octobre dernier le microdrive était présenté à la presse informatique française par Nigel Searle, le Directeur général de Sinclair Research. A cette occasion Direco International, le représentant exclusif en France de la firme britannique, en la personne de son président Eric Bompard, a informé les journalistes présents que 132000 micro ordinateurs familiaux ZX avaient été vendus. Ceci correspond d'après lui à plus de la moitié du marché français dans l'informatique familiale. Une autre information intéressante : une opération pourrait être montée visant à doter une centaine de lycées et collèges de machines ZX.

La société Sinclair pèse à ce jour 600 millions de francs de chiffre d'affaire ce qui est 2 fois plus qu'en 1982.

EXPRESSION ECRITE

L'imprimante Sinclair est encore, 2 ans après sa conception parfaitement dans la course pour le rapport qualité/prix ; elle est si bon marché que de nombreux autres ordinateurs aimeraient bien pouvoir se l'adjoindre. C'est possible maintenant, du moins pour les ordinateurs construits autour du microprocesseurs 6502 grâce à une interface proposée par Microtonic Computers systèmes LTD pour environ 350 F.

Maintenant que l'interface RS 232 est disponible pour le Spectrum il va être possible de regarder du côté des imprimantes un peu plus sophistiquées que celle du constructeur sans rechercher pour autant des interfaces encombrantes en provenance de fournisseurs plus ou moins réputés. Il existe chez Brothers une jolie petite imprimante ultra plate l'EP 22 avec interface incorporée qui coûte en France moins de 3000 F. Nous essaierons de savoir si elle est compatible avec le Spectrum.

Un traitement de texte professionnel pour Spectrum 48 K avec justification à droite et 64 caractères par ligne ? C'est ce que propose le groupe Sémaphore, consultants en communications. Naturellement ce logiciel est compatible avec la majorité des imprimantes professionnelles.

Pour ceux qui voudraient utiliser leur Spectrum directement interfacé à une imprimante de qualité sans attendre la commercialisation de l'interface 1 sur le continent, il est possible de se procurer chez KEMPSTON Microelectronics pour le prix de 500 F une interface RS 232 munie de son câble : cette interface à la taille d'une extension mémoire 16K de ZX81.

COMBIEN LA PUCE ?

Bientôt l'octet de mémoire vive à un centime ? La chose est désormais en vue avec les nouvelles additions mémoire pour ZX81 qui sont commercialisées en Angleterre pour moins de 16 livres, par Microcare.

La différence financière entre un Spectrum 16K et un Spectrum 48K est en passe de devenir négligeable : Fox electronics propose un kit avec une dizaine de puces à fixer soi-même à l'intérieur du Spectrum pour 20 livres anglaises.

AUX QUATRE Q.I. DU MONDE

Un ZX parlant arabe ? Cela existe, c'est un négociant de Jeddah qui le commercialise en Arabie Séoudite : les modifications de la MEM ont dû être radicales ; d'une part en Arabe les lettres n'ont pas le même graphisme selon qu'elles se trouvent au début, au milieu ou la fin des mots et ensuite et surtout, ce que personne n'ignore, l'Arabe s'écrit de droite à gauche, l'ensemble des fonctions d'éditions est donc revu.

Sinclair va fournir des ZX81 et des Spectrums en Kits à la South China Computer Company (GUANGZHOU-Canton). Les premiers engagements portent sur un marché de pres de 10 millions de livres.

CLUBS

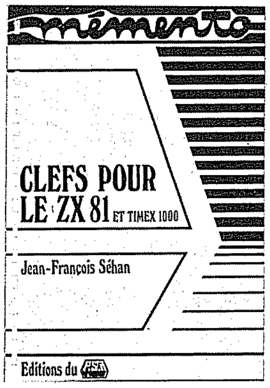
- Microtel ADEMIR TREMBLAY, Collège Romain Rolland, Rue de Reims, 93410 VAUJOURS (Sec. Pascal VIEVILLE).
- ZX Microclub Résidence Europa, 6 Bd Léopold III, B-1030 BRUXELLES (Sec. Michel Hunin).
- Club Informatique de Conflans, College Jules Ferry, CONFLANS (Sec. Mme Mlet).

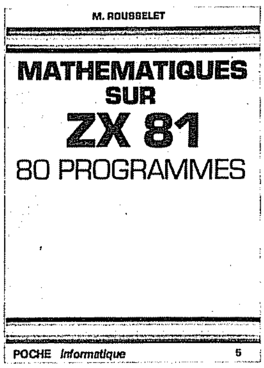
ADRESSES DES SOCIETES MENTIONNEES

- ARAB RAM, RAMEZ HALABY & CO, P.O. BOX 147, JEDDAH, Arabie Séoudite.
- MICROCARE, 79 Thane Works, Thane Villas, LONDON N7, GB.
- FOX ELECTRONICS, 141 Abbey Road, Basingstoke, HANTS, GB.
- AMERSHAM SOFTWARE LTD, Long Roof, Hervines Rd, Amersham, BUCKS, GB.
- MEMOTECH, Memotech Limited, Station Lane Industrial Estate, Witney, Oxon, 078 6BX, GB.
- SEMAPHORE, JP Cardinaux, CH 1283 LA PLAINE (GENEVE) Suisse.
- E.G. de la M.INFO, EGNI, 20 Bd St Denis, 75010 PARIS.
- INNELEC, 110 bis Av. du Général Leclerc, 93500 PANTIN.
- SILICONE VALLEY, 87 quai de la Fosse, NANTES
  7 rue Boisnet, ANGERS
- MICROTANIC COMPUTERS SYSTEMS, 16 Usland Rd, Dulwich, LONDON SE 22, GB.
- KEMPSTON Microelectronics, Unit 30, Singer Way, Woburn Road, I.E., KEMPSTON, Bedfort NK 42 7AF, GB.

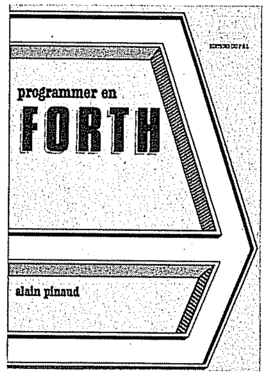
Librairie











PASSEPORT POUR ZX81

Par Claudy Galais
Collection Poche Informatique
Prix : 50 F environ
128 pages

Sixième titre de la toute jeune collection "POCHE Informatique ETSF", PASSEPORT POUR ZX81 doit avant tout être considéré comme un ouvrage de référence. Il s'adresse à notre avis surtout aux habitués d'autres machines souhaitant se familiariser rapidement avec les particularités du BASIC SINCLAIR.

Vu sous cet angle, le classement alphabétique des mots-clé se justifie par un souci de commodité d'emploi, alors qu'il aurait constitué une faute de conception dans un ouvrage de pure initiation, dont l'approche doit être très progressive. Chaque mot-clé est illustré par un petit programme d'une originalité certaine : généralement dénué de toute utilité pratique, il est conçu de manière à utiliser le mot-clé étudié, dans tous les cas susceptibles de se présenter.
Toutes les variantes de mise en oeuvre étant ainsi passées en revue, les surprises ultérieures se trouvent éliminées de façon élégante.

Le débutant ayant appris le BASIC au moyen du ZX81 trouvera dans PASSEPORT, grâce à sa présentation "index", les réponses à ses petits problèmes bien plus rapidement qu'en consultant le manuel.

Une seule contrainte, mais impérative : les programmes fournis doivent être scrupuleusement frappés et exécutés.
C'est en la voyant à l'oeuvre que l'on comprend le fonctionnement d'une instruction, même simple. Parcourir le listing, même avec application, ne suffit pas, car certaines réactions de la machine se trouvent escamotées.

Comment deviner, par exemple, que le curseur d'une instruction INPUT apparaît entre guillemets lorsqu' une chaîne est attendue, mais tout seul s'il s'agit d'une valeur numérique ? PASSEPORT POUR ZX81 est un bon outil de travail, mais seulement un outil. Il appartient à son lecteur de s'en servir avec application pour en tirer le maximum de profit.

P. G.

CLEFS POUR LE ZX81 ET TIMEX 1000

Par Jean-François Sehan
Editions du PSI
Prix : 82 F

Il suffisait d'y penser ! Telle est en effet la conclusion qui se présente immédiatement à l'esprit au terme de la lecture de cet ouvrage "pas comme les autres".
Aide-mémoire, mémento, bloc-notes, ne sont que quelques exemples de termes pouvant servir à définir ce petit livre dont même la reliure à anneaux semble inviter à une consultation de tous les instants.

On ne trouvera guère d'inédit dans CLEFS POUR LE ZX81, qui se présente un peu comme la "mise au propre" de toute la documentation qu'un passionné du ZX aurait pu glaner ici ou là en un an ou deux de recherches. Certaines rubriques proviennent en droite ligne du manuel SINCLAIR (jeu de caractères, variables-système, index des mots-clé, codes d'erreur, etc.), mais J.F. Sehan a toujours tenu à les compléter par des commentaires d'inspiration personnelle parfois fort judicieux.
Le langage machine Z80 occupe un chapitre entier, avec divers tableaux permettant l'assemblage ou le désassemblage à la main dans les meilleures conditions.
On pourra regretter l'utilisation exclusive de la notation hexadécimale, mais fort heureusement un très pratique tableau de conversion permet un retour immédiat au code décimal, seul compris par le petit Sinclair.

On appréciera à sa juste valeur le recueil d'adresses commentées des routines de la ROM, qui ne pouvait guère être consulté, jusqu'à présent, que dans le rare et coûteux livret de Ian Logan, dont l'auteur a fort probablement pris le risque de s'inspirer très largement...

Après le logiciel, le matériel : le schéma du ZX81 n'étant fourni qu'aux acheteurs de la version "kit", les autres seront heureux de trouver un plan simplifié (peut-être un peu trop !), et des informations détaillées sur les brochages des circuits intégrés et du connecteur arrière.

Bien qu'utiles, ces données techniques ne suffiront pas aux risque-tout désireux d'intervenir à l'intérieur même de la machine, et qui devront se procurer

ailleurs le schéma complet, qui aurait indiscutablement été à sa place parmi cette abondance de documentation.

En guise de conclusion, un chapitre "TRCUS ET ASTUCES" founit une sélection de tours de main qui nous rappellent pour la plupart des lectures antérieures...

Les grandes qualités de ce livre sont celles de la collection où il figure. Le mérite personnel de J.F. Sehan est d'avoir regroupé de façon cohérente un grand nombre d'informations puisées aux sources les plus diverses : ce petit livre de cent cinquante grammes en dit autant sur le ZX81 qu'une quinzaine de kilos de documentation brute.

Beau travail de synthèse donc, malgré une nette faiblesse du côté des données relatives au circuit interne de la machine. Quoi qu'il en soit, CLEFS POUR LE ZX81 rendra de fiers services à l'utilisateur "sérieux" du ZX81, en lui évitant de perdre un temps précieux à compulser pour un point précis une documentation personnelle plus éparpillée.

P. G.

MATHEMATIQUES SUR ZX81

Par Michel Rousselet
Collection Poche Informatique
Prix : 50 F environ
128 pages

Les Fonctions mathématiques du ZX81, très riches, ne servent le plus souvent qu'à la construction de décors pour les jeux vidéo les plus divers.
Cela est bien dommage, car une telle puissance de calcul aurait fait la joie des étudiants des années soixante-dix, peinant sur d'antiques engins à réglette et curseur coulissants...

Il n'est cependant pas trop tard pour réagir, comme en témoigne cet intéressant petit livre, qui s'adresse sans nul doute à un public estudiantin ou scientifique.
Dès le troisième chapitre, le niveau du discours rejoint celui de la classe de "Math'sup" pour dépasser allègrement celui de "Math'spé" vers la fin du volume !

Nous étions bien loin de soupçonner, avant de lire cet ouvrage, que le ZX81 pouvait résoudre des équations différentielles, intégrer des fonctions numériques par les méthodes de Simpson et Tchebytchev (pas moins...), et faire merveille en matière de probabilités et statistiques.

Tout cela non pas par simple amour de l'Art, car les programmes proposés peuvent rendre de fiers services dans bien des disciplines scientifiques, même de haut niveau, soit directement, soit par incorporation dans des logiciels plus importants.

Ce livre original est un cadeau absolument idéal (et peu coûteux, puisqu'édité en collection de poche) à faire à un lycéen enclin à passer un peu trop de temps devant son clavier, au point parfois de négliger ses révisions...
Bien que l'usage d'un ZX81 ne soit pas encore autorisé en salle d'examen, gageons que le temps passé devant un écran sur lequel se résoud une équation diophantienne (mais oui !), sera plus profitable qu'une partie d'astéroïdes ou d'envahisseurs.

L'informatique individuelle risque de décevoir certains de ses adeptes par manque d'applications pratiques. MATHEMATIQUES SUR ZX81 constitue un remède efficace contre ce mal naissant, applicable d'ailleurs à d'autre machines programmables en BASIC, et notamment au SPECTRUM.

P. G.

SPECTRUM HARDWARE MANUAL
Par Adrian Dickens
Editeur : MELBOURNE HOUSE
Prix : 120 F environ
150 pages

Voici un livre qui s'adresse particulièrement aux électroniciens. Il contient, en effet, le schéma complet du ZX Spectrum, la description technique des composants et des projets de périphériques.

Dans les premiers chapitres, l'auteur passe en revue tous les composants discrets (alimentation, circuit vidéo, circuit H.P.) et monolithiques (microprocesseur, mémoire morte, etc...), en précisant la fonction de tous les signaux présents à chaque broche des boîtiers. Nous apprenons ainsi que l'alimentation d'un Spectrum 16 K peut supporter d'alimenter des circuits extérieurs jusqu'à concurrence de 300 mA, mais qu'un Spectrum de 48 K ne le peut pas. On nous explique également que la mémoire vive de base de 16 K peut être employée par le microprocesseur pour dérouler un programme BASIC, ou pour manipuler les variables système, et également, par le circuit spécialisé Sinclair C.I. 1 pour réaliser l'affichage. Cette demande de mémoire par 2 circuits différents crée un conflit que le circuit spécialisé Sinclair règle en suspendant l'horloge du microprocesseur pendant quelques cycles. On nous dit que les signaux vidéo n'ont pas été raccordés au connecteur arrière. On nous explique les différences entre les différentes versions du ZX Spectrum, et beaucoup d'autres détails. Chaque chapitre débute par la photographie (vue côté composants) du Spectrum, version 1 et version 2, avec encadrement des éléments analysés. Cela fait un bon nombre de pages identiques, d'autant plus inutiles qu'en appendice on trouve l'emplacement des composants pour chaque version. Nous avons trouvé insuffisants les renseignements nécessaires pour transformer un Spectrum 16 K en un 48 K ; des éléments pour le réaliser sont cependant présents dans le livre.

Dans les derniers chapitres (un gros tiers de l'ouvrage), l'auteur donne des éléments pour l'addition de périphériques, par l'intermédiaire du connecteur arrière. Après avoir expérimenté quelques circuits simples (entre autres, un bouton RESET, bien utile quand l'ordinateur se plante : on n'a plus besoin de couper puis de reconnecter l'alim.), l'auteur décrit quelques réalisations : un PIO qui estune interface parallèle d'E/S avec lequel on dispose de 2 ports de 8 bits chacun ; une interface pour raccorder un 2ème clavier alphanumérique ou un clavier hexadécimal ou tout arrangement de touches souhaité ; un ou deux joysticks pour la commande de mouvements avec un programme Apollo comme exemple et enfin, un convertisseur analogique/digital avec un programme pour dessiner sur l'écran comme exemple d'utilisation. Le fait que ces réalisations sont accompagnées de schémas de principe, de schémas de câblage et de nomenclatures les rendent accessibles aux bricoleurs éclairés, cible directe de ce livre soigné.

Y. D.

PROGRAMMER EN FORTH

Par Alain Pinaud
Editions du PSI
Prix : 82 F environ
160 pages

Le FORTH est en train de susciter un intérêt croissant et ORDI-5 a des raisons particulières de s'intéresser à lui : c'est en effet la langue "vernaculaire" d'un ordinateur très sympathique que nous avons décidé d'adopter dans notre revue, le Jupiter Ace, Désormais tous les ordinateurs Sinclair peuvent aussi être implementés en FORTH grâce à des cassettes supportant ce langage : il y a même désormais plusieurs versions par ordinateur, produite chacune par une maison de soft réputée outre manche.

Seulement voilà : existe-t-il un livre simple pour apprendre ce langage somme toute difficile ? Tant qu' il était nécessaire de se tourner vers la documentation anglo-américaine pour approfondir ses connaissances dans ce domaine la réponse était "non" ; depuis la parution chez PSI du livre de Alain Pinaud : "programmer en forth" la réponse est péremptoirement "oui".

Chaque auteur a son truc : celui d'Alain Pinaud est de savoir présenter des choses difficiles dans un style enjoué et qui passe très bien ; nous avions déjà beaucoup apprécié son livre "programmer en assembleur" grâce auquel plus d'un possesseur de ZX81 a osé s'aventurer dans le domaine du microprocesseur Z80 ; pourtant l'assembleur Z80 n'est pas chose aisée à présenter : on compte sur les doigts les bons livres qui peuvent en parler simplement. Les mêmes qualités se retrouvent dans cet ouvrage : l'humour le dispute à la compétence. L'approche est progressive et assez gratifiante à condition d'avoir son Jupiter Ace ou son TRS80 sous la main constamment, et de bons résumés font le point régulièrement, sans négliger les exercices qui constituent l'un des points forts de la collection.

Pour tous ceux qui possèdent le Jupiter Ace et qui trouvent que le manuel qui l'accompagne pèche souvent par élision, un second livre est absolument nécessaire et nous conseillons celui-là. En termes d'initiation, cet ouvrage est pratiquement sans défauts. Une chose manque cependant absolument et nous voudrions bien que dans une seconde édition il y soit porté remède : l'absence d'un index analytique se fait cruellement sentir et rend le livre, une fois lu, difficile à manier et remanier au titre de la référence. Mais il est très probable que le "fortheur" qui aura traversé ce livre avec profit envisagera l'acquisition d'un autre ouvrage plus théorique et plus systématique. Cet ouvrage pourrait s'intituler très justement le "forth sans peine" si la vénérable collection que chacun connait s'ouvrait un jour aux langages de programmation.

X. L. B.

Trucs à Brac

LE COIN DES INDISCRETS
(POUR APPRENDRE SE MEFIER DES TRUCS DES PIRATES)

Il est souvent intéressant de LISTER un programme pour en comprendre le fonctionnement, les diverses astuces et trucs, ou encore tout simplement pour le modifier ; malheureusement la plupart des programmes vendus dans le commerce sont en langage machine (ce qui empêche l'utilisateur d'arrêter le programme en agissant sur la commande BREAK) et s'exécutent automatiquement après leur chargement par la commande LOAD "nom du programme".

Voici une petite routine toute simple, à entrer directement au clavier, qui permet de palier à cet inconvénient et qui peut encore rendre d'énormes services (à condition de ne pas l'utiliser pour une duplication abusive de logiciels), par exemple : la traduction française de jeu en version anglaise.

Déroulement :

1) Positionner le micro-ordinateur en mode FAST ;
2) Taper RAND USR 836 suivit de NEWLINE ;
3) Appuyer sur PLAY comme lors d'un LOAD normal ;
4) Attendre la fin du chargement : un compte-rendu du style C/... apparaît, vous pouvez LISTER votre programme.

Voilà, vous avez vu ce n'est vraiment pas compliqué, et ça rend souvent d'énormes services (par exemple : obtenir le LISTING d'un programme qui se charge automatiquement au-dessus de RAMTOP).

N.B. Une petite explication tout de même concernant le RAND USR 836, il faut savoir que dans la ROM Basic de Sinclair, il existe un micro-programme correspondant à la commande LOAD à l'adresse décimale 832 (en hexadécima :3C2), c'est donc une partie de cette routine que l'on utilise.

Philippe Buschini

II - Entrer et faire tourner le programme I puis effacer les lignes 10 à 50.
Taper ensuite les lignes 10 et 15 du programme II que l'on sauvegarde en faisant GOTO 10. C'est le programme qu'il faudra introduire dans le ZX au lieu de taper LOAD"" quand on voudra lister un programme protégé.

```
  1 REM 12345678
 10 LET A=16514
 15 POKE A,205
 20 POKE A+1,35
 25 POKE A+2,15
 30 POKE A+3,203
 35 POKE A+4,250
 40 POKE A+5,195
 45 POKE A+6,71
 50 POKE A+7,3

  1 REM LN ??ACS IF ??
 10 SAVE "PIRATAGE"
 15 PRINT "  FAITES ""RAND USR
16514"" PUIS METTEZ VOTRE MAG
NETOPHONE EN POSITION ""LECTURE
"""
```

François Hache

TRUC A BREAK

Comment faire en sorte que la touche BREAK soit désensibilisée de façon à interdire le piratage de ses programmes Spectrum ? Il suffit de faire POKE 23659,0.

SAVE LOAD (ZX81), UNE AMELIORATION DECISIVE !

La solution que j'apporte n'a, à ma connaissance, jamais été publiée malgré les nombreux articles sur le sujet.
Elle est pourtant d'une grande simplicité. Il faut intercaler, dans la liaison MIC, une résistance dont la valeur optimum est à déterminer expérimentalement et dépent essentiellement de la bonne entente des deux protagonistes.

Deux exemples précis permettent de mieux apprécier :
- 1er cas : impossibilité absolue de réaliser un enregistrement fiable.
A la lecture, avant même l'arrivée des signaux caractéristiques, l'écran devient blanc, le ZX renvoie le curseur K.
Le ZX étant hors de cause, la partie lecture également (aucun problème pour lire une cassette pré-enregistrée pro. ou amateur) c'est bien la partie enregistrement qui est en cause. L'insertion d'une résistance de 6800 Ohms (la tolérance est assez grande) permet de réaliser des enregistrements fiables, à 90 %, en vitesse normale.
Par contre, dans ce cas-là, aucun succès avec le FAST LOAD MONITOR (FLM).
D'ailleurs, la lecture est, à ces vitesses, en cause car il est impossible de lire une cassette pré-enregistrée ; je n'ai pour l'instant aucune solution à proposer, les réglages d'Azimut sont sans effet.

- 2ème cas : aucun problème d'enregistrement à vitesse normale. Fiabilité 100 %.
Par contre en FLM aucun enregistrement n'est lisible ; la fonction 3 (titre) donne des résultats étranges : les titres affichés ressemblent au REM du code machine, ce qui me guide sur la bonne voie, le début de l'enregistrement est saturé, déformé.
L'insertion d'un potentiomètre de précision de 20 kohm me permet de constater la progression des résultats.
L'augmentation progressive de la résistance série dans la liaison MIC permet de voir les titres se transformer d'abord en "AST LOAD MONITOR" puis en titres réels à la vitesse 9.
En descendant l'échelle des vitesses, on constate à nouveau un déraillement dans la reproduction du titre : augmentation de la résistance et de nouveau de bons titres.

MODIFICATION DE RAMTOP PAR UN PROGRAMME

Bien des programmes, et notamment des jeux du commerce, nécessitent la réservation de place mémoire au-dessus de RAMTOP.
Il est alors d'usage que la notice invite l'utilisateur à exécuter un ou deux POKE en mode commande, puis un NEW. Ces préparatifs fastidieux et générateurs de

Le LOAD de la ligne 40 exécutera l'équivalent d'un NEW et chargera immédiatement le programme utile.
Le SAVE de la ligne 10 permet d'enregistrer le programme amorcé par un simple RUN, tout en lui permettant de se lancer seul lors du rechargement.
Ce procédé est ainsi entièrement "transparent" à l'utilisateur qui n'aura aucune consigne particulière à suivre pour charger la cassette par un simple LOAD"".
Il sera avantageux d'enregistrer le programme amorcé sans extension mémoire en place : on accélèrera ainsi risques d'erreurs peuvent être facilement éliminés par un chargement en deux temps du logiciel.
Il n'est pas possible, en effet, de placer ces instructions en tête du programme principal, puisque le NEW nécessaire effacerait toute la partie utile du logiciel !
On peut par contre loger sur la cassette, juste avant le programme principal, les quelques lignes suivantes :

```
10 SAVE "AMORCE"
20 POKE 16388, octet bas de Ramtop
30 POKE 16389, octet haut de Ramtop
40 LOAD ""
```

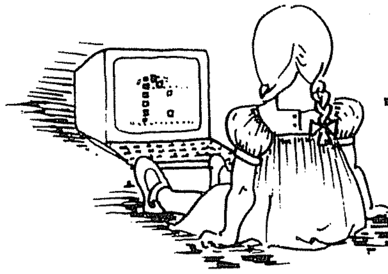
notablement les opérations.
Enfin, on peut songer à ajouter les lignes suivantes, qui permettent un rapide contrôle du réglage du magnétophone :

```
32 PRINT AT 10,0; "NIVEAU CORRECT"
34 PRINT AT 15,0; "ATTENDEZ SVP"
36 PAUSE 200
```

Patrick Gueulle

pour vos enfants

Juvénilia



Michel Murbach



Le Schtroumpf programmeur

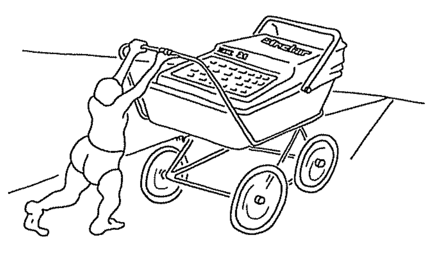
Qu'est-ce que la rubrique "Juvénilia" ?
C'est le point de rencontre des oeuvres de jeunesse, comme disent les latinistes en parlant des grands auteurs classiques. La rédaction d'ORDI-5 a dans l'idée que sur les quelque 50 000 utilisateurs francophones du ZX81, il doit y avoir une très forte proportion de jeunes et même de très jeunes.
Vous qui avez moins de 14 ans, cette rubrique vous est ouverte. Bien sûr, si vous êtes déjà capable, comme le jeune Simon LANE, de nous écrire un programme de 800 instructions pour la résolution du Rubik's Cube, on vous trouvera une autre place dans la revue !

Juvénilia 1K

```
 1 LET N=CODE "■"
 2 LET D=CODE ""
 3 INPUT H
 4 LET Q=D
 5 LET X=N
 6 LET Y=X+CODE " ■"
 7 LET T=N
 8 GOTO 55
10 PRINT AT N,X;"*";TAB X+H;"*
19 LET T=T+CODE "■ "
20 LET D=D+(RND-RND)/CODE " ■"
21 IF ABS D>X/X THEN LET D=X-X
30 LET X=X+SGN D
40 IF NOT X THEN LET X=Y/Y
41 IF X>CODE "=" THEN LET X=20
44 SCROLL
45 PRINT AT CODE "",Y-1;"■▒■"
49 LET A$=INKEY$
50 IF X>CODE "■" AND X<>CODE "
=" THEN LET K=K+SGN D
52 IF A$="5" THEN LET Y=Y-X/X
53 IF A$="8" THEN LET Y=Y+X/X
54 IF T/N=INT (T/N) AND (Y<R+K
OR Y>H+R+K) THEN GOTO 90
55 IF T/N=INT (T/N) THEN LET K
=CODE ""
56 IF T/N=INT (T/N) THEN LET R
=X
70 GOTO CODE "▒"
80 PRINT T,Q
91 IF T>Q THEN LET Q=T
92 GOTO CODE "■"
```

Bruno Célerier

Rallye



Zedix ou l'ordinateur expliqué par lui-même

Ce beau programme éducatif de Claude Remy affiche un éclaté du ZX avec ses différents circuits intégrés et les liaisons qui les relient entre eux et simule d'une façon assez spectaculaire le cheminement des impulsions de courant et les modifications d'état des cellules de mémoire. Après cela le ZX ne vous paraîtra pas plus compliqué qu'un beau boulier, même si vous avez un peu tâtonné au début.

LA PLANCHE D'ANATOMIE DU ZX : LA MEMOIRE, L'UNITE DE TRAITEMENT ET LES LIAISONS

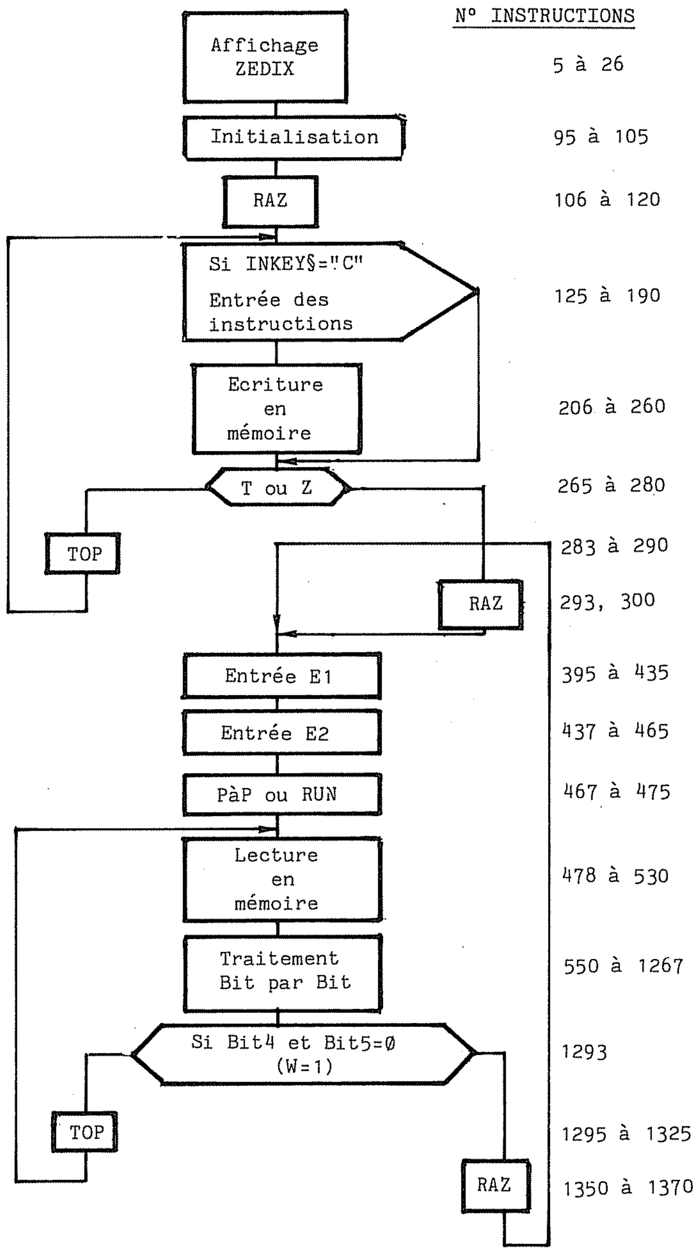
Le ZX est un ordinateur très simple mais qui possède les organes essentiels des grands systèmes : une mémoire et une unité de traitement.
Comme tout ordinateur, il ne connaît que 2 états :

- le courant électrique passe : bit 1 : ************
- le courant ne passe pas : bit Ø : ....

* La mémoire permet de stocker 64 bits ou 8 mots de 8 bits. Chaque mot occupe une rangée. Chaque emplacement de bit est représenté par un point.
Les mots entrent ou sortent de la mémoire par 8 fils raccordés au bus dont les 8 fils ont été réunis en un gros trait.
Pour accéder à une rangée (pour y écrire ou pour lire un mot) il faut que la ligne-mémoire correspondante, qui sort du décodeur, soit à l'état 1.
Les lignes-mémoire sont successivement à l'état 1 au rythme des tops délivrés par une horloge (non représentée sur l'écran).
Les tops (bref passage de courant électrique) sont comptés par un compteur qui délivre un nombre binaire correspondant au nombre de tops.
Ce nombre binaire constitue l'adresse de la mémoire.
Elle est décodée par le décodeur qui, à chaque nombre binaire, fait correspondre dans l'ordre, une ligne-mémoire.
Pour comprendre le principe de représentation à l'écran :

. Initialiser le système par une remise à éro (Raz : Touche Z)
. Envoyer successivement des tops (Touche T) - (taper C (comme Compteur) quand l'écran demande Ø ou 1.

	Compteur Système décimal	Nb binaire =adresse	Ligne-Mémoire
Raz	0	ØØØ	1
1er Top	1	ØØ1	2
2e Top	2	Ø1Ø	3
3è Top	3	Ø11	4
	... ETC ...		
7e Top	7	111	8



* L'unité de traitement n'opère que sur des entrées à un bit (Ø ou 1).
Les données sont introduites en E1 et E2.
Le traitement consiste à résoudre les fonctions logiques de base : ET, OU, PAS :

- Fonction ET : le résultat = 1 si les 2 entrées sont égales à 1.
- Fonction OU : le résultat = 1 si une entrée au moins est égale à 1.
- Fonction PAS : (n'opère que sur une seule entrée) : le résultat = 1 si l'entrée est égale à Ø.

L'opération logique est effectuée par l'unité de calcul LOG.
Le résultat apparaît dans les cases R1 ou R2. Ce résultat peut être réintroduit par les entrées E3 et E4.
Chaque entrée est validée par une porte ET "*", qui laisse passer le bit de l'entrée si la ligne venant du bus est à l'état 1. Il en est de même pour les résultats issus de LOG.
L'opération effectuée par l'unité de calcul LOG dépend de l'état des lignes 4 et 5 issues du bus, selon le code suivant :

Ligne 4	Ligne 5	Opération effectuée
Ø	Ø	Arrêt
Ø	1	Opération ET
1	Ø	Opération OU
1	1	Opération PAS

On peut maintenant déterminer les instructions à écrire en mémoire et qui commanderont l'unité de traitement :

1er exemple :

Opération ET entre E1 et E2, résultat en R1

	Validation des entrées				Code opératoire		Validation de la sortie	
N° de la ligne de bus	0	1	2	3	4	5	6	7
E1 ET E2 = R1	1	1	Ø	Ø	Ø	1	Ø	1

2ème exemple :

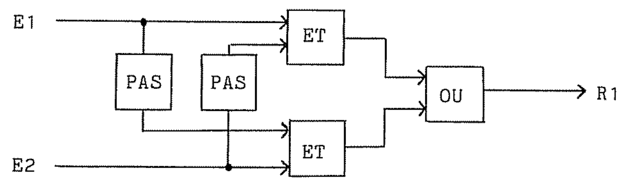
Opération PAS sur l'entrée E3, résultat en R2
PAS E3 = R2 : ØØ1Ø11Ø1 ou 2D en hexadécimal

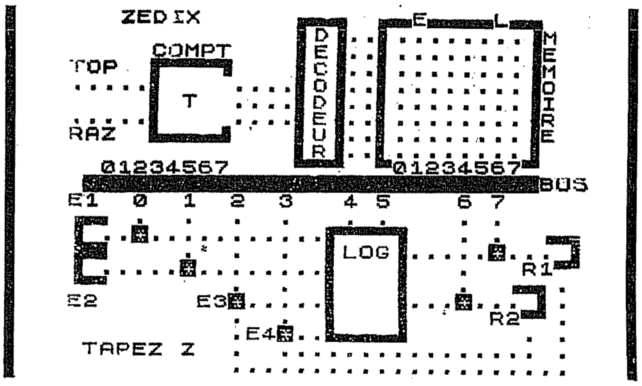
Il faut commencer par stocker les instructions en mémoire.
Pour écrire un mot en mémoire, il faut :
- qu'un mot soit présent sur le bus. Les mots-instructions sont introduits par l'organe d'entrée 0 à 7 à gauche du bus.
- que la ligne-mémoire correspondant à la rangée désirée, soit à l'état 1 et par conséquent qu'une adresse soit appliquée au décodeur.
- que l'ordre E (écriture) soit à l'état 1.

De même, pour lire un mot en mémoire, il faudra :
- que la ligne-mémoire correspondante soit à l'état 1.
- que l'ordre L (lecture) soit à l'état 1.

VISUALISEZ UN EXEMPLE DE TRAITEMENT

Essayons de résoudre la fonction OUX (OU exclusif) :
R1 = 1 si une seule des entrées E1 ou E2 = 1






```
   1 REM *ZEDIX*
   2 REM (C)CL.REMY ET ORDI-5
   3 REM ------------------
   5 PRINT "   ZEDIX          ▀▀▀▀  ▀E▀▀▀▀L▀▀"
   6 PRINT TAB 14;"█D█..█........█M"
   7 PRINT "         COMPT    █E█..█........█E"
   8 PRINT "TOP  ▀▀▀▀▀▀█    █C█..█........█M"
   9 PRINT ".....█      ....█O█..█........█O"
  10 PRINT "     █ T    ....█D█..█........█I"
  11 PRINT ".....█      ....█E█..█........█R"
  12 PRINT "RAZ  ▄▄▄▄▄▄█    █U█..█........█E"
  13 PRINT TAB 14;"█R█..█........█"
  14 PRINT "  01234567    ▀▀▀▀  ▀01234567▀▀"
  15 PRINT " ████████████████████████BUS"
  16 PRINT "E1  0  1  2  3   4 5  6 7"
  17 PRINT "▀▀▀  .  .  .  .   . ."
  18 PRINT "█ ..█...........█▀▀▀▀▀▀█"
  19 PRINT "▀▀▀  .  .  .   █LOG █....█...  █"
  20 PRINT "█ .....█........█  █ R1▀▀"
  21 PRINT "▀▀▀     .  .   █  █ ▀▀ ."
  22 PRINT "E2     E3█.....█  █...█...  █."
  23 PRINT TAB 10;".  .  █     █  R2▀▀ ."
  24 PRINT TAB 10;".E4█..▄▄▄▄▄▄█"
  25 PRINT TAB 10;".  ..........  ........."
  26 PRINT TAB 10;"....................."
  95 DIM M(8,8)
 100 LET K=0
 105 LET L=0
 108 PRINT AT 20,1;"TAPEZ Z"
 110 IF INKEY$="Z" THEN GOTO 118
 115 GOTO 110
 118 GOSUB 9000
 120 GOSUB 5000
 125 PRINT AT 20,1;"0 OU 1 ?"
 126 LET C=0
 127 LET L=T+1
 130 IF INKEY$<>"" THEN GOTO 130
 135 IF INKEY$="" THEN GOTO 135
 140 LET M$=INKEY$
 145 IF M$="C" THEN GOTO 265
 150 LET C=C+1
 155 LET M(L,C)=VAL M$
 156 PRINT AT 21,C;M(L,C)
 160 IF M$="0" THEN PRINT AT 9,(C+1);CHR$ (27+C);TAB (C+19);CHR$ (27+C)
 180 IF M$="1" THEN PRINT AT 9,(C+1);CHR$ (155+C);TAB (C+19);CHR$ (155+C)
 185 IF C=8 THEN GOTO 206
 190 GOTO 130
 206 PRINT AT 21,1;"            "
 207 PRINT AT 20,1;"TAPEZ E "
 210 IF INKEY$="E" THEN GOTO 218
 215 GOTO 210
 218 GOSUB 9000
 220 PRINT AT 0,21;"█"
 230 FOR C=1 TO 8
 235 IF M(L,C)=0 THEN GOTO 250
 240 PRINT AT L,(19+C);"█"
 245 GOTO 255
 250 PRINT AT L,(19+C);"."
 255 NEXT C
 260 PRINT AT 0,21;"E"
 265 PRINT AT 20,1;"T OU Z  "
 270 IF INKEY$="Z" THEN GOTO 293
 275 IF INKEY$="T" THEN GOTO 283
 280 GOTO 270
 283 GOSUB 9000
 285 GOSUB 6000
 290 GOTO 125
 293 GOSUB 9000
 300 GOSUB 5000
 395 PRINT AT 9,2;"                "
 398 LET V=0
 400 LET S=0
 405 LET W=0
 407 PRINT AT 20,1;"E1 ?      "
 409 IF INKEY$="0" THEN GOTO 412
 410 IF INKEY$="1" THEN GOTO 414
 411 GOTO 409
 412 LET D=0
 413 GOTO 415
```

```
414 LET D=1
415 PRINT AT 13,1;D
420 IF D=0 THEN GOTO 435
425 PRINT AT 13,2;"**"
430 GOTO 437
435 PRINT AT 13,2;".."
437 PRINT AT 20,1;"E2 ?"
439 IF INKEY$="0" THEN GOTO 442
440 IF INKEY$="1" THEN GOTO 444
441 GOTO 439
442 LET E=0
443 GOTO 445
444 LET E=1
445 PRINT AT 15,1;E
450 IF E=0 THEN GOTO 465
455 PRINT AT 15,2;"*****"
460 GOTO 467
465 PRINT AT 15,2;"....."
467 PRINT AT 19,1;"P A P :P"
468 PRINT AT 20,1;"OU RUN:R"
470 IF INKEY$<>"" THEN GOTO 470
471 IF INKEY$="" THEN GOTO 471
472 LET K$=INKEY$
473 GOSUB 9000
475 IF K$="R" THEN GOTO 488
478 PRINT AT 20,1;"TAPEZ L "
480 IF INKEY$="L" THEN GOTO 488
485 GOTO 480
488 GOSUB 9000
490 PRINT AT 0,26;"■"
495 LET L=T+1
500 FOR C=1 TO 6
505 IF M(L,C)=0 THEN GOTO 520
510 PRINT AT 9,(19+C);CHR$ (155
+C)
515 GOTO 525
520 PRINT AT 9,(19+C);CHR$ (27+
C)
525 NEXT C
530 PRINT AT 0,26;"L"
550 IF M(L,1)=0 THEN GOTO 585
555 PRINT AT 11,4;"■"
560 PRINT TAB 4;"*"
565 IF D=0 THEN GOTO 595
570 LET N=1
575 PRINT TAB 5;"************"
580 GOTO 650
585 PRINT AT 11,4;"0"
590 PRINT TAB 4;"."
595 LET N=0
600 PRINT TAB 5;"............"
650 IF M(L,2)=0 THEN GOTO 685
655 PRINT AT 11,7;"■"
660 FOR X=12 TO 14
661 PRINT AT X,7;"*"
662 NEXT X
665 IF E=0 THEN GOTO 695
670 LET O=1
675 PRINT AT 15,8;"*********"
680 GOTO 750
685 PRINT AT 11,7;"1"
690 PRINT AT 12,7;"."
691 PRINT AT 14,7;"."
695 LET O=0
700 PRINT AT 15,8;"........."
750 IF M(L,3)=0 THEN GOTO 785
755 PRINT AT 11,10;"■"
760 FOR X=12 TO 16
761 PRINT AT X,10;"*"
762 NEXT X
765 IF V=0 THEN GOTO 795
770 LET P=1
775 PRINT AT 17,11;"*****"
780 GOTO 850
785 PRINT AT 11,10;"2"
790 FOR X=12 TO 16 STEP 2
791 PRINT AT X,10;"."
792 NEXT X
795 LET P=0
800 PRINT AT 17,11;"....."
850 IF M(L,4)=0 THEN GOTO 885
855 PRINT AT 11,13;"■"
860 FOR X=12 TO 16
861 PRINT AT X,13;"*"
862 NEXT X
865 IF S=0 THEN GOTO 895
870 LET Q=1
875 PRINT AT 19,14;"**"
880 GOTO 950
885 PRINT AT 11,13;"3"
890 FOR X=12 TO 16 STEP 2
891 PRINT AT X;13;"."
892■NEXT X
895 LET Q=0
900 PRINT AT 19,14;".."
1000 IF M(L,5)=0 THEN GOTO 1038
1005 PRINT AT 11,17;"■"
1006 PRINT TAB 17;"*"
1010 IF M(L,6)=0 THEN GOTO 1030
1015 PRINT AT 11,19;"■"
1016 PRINT AT 12,19;"*"
1019 PRINT AT 16,17;"PAS"
1020 LET U=1
1021 IF N=1 OR O=1 OR P=1 OR Q=1
 THEN LET U=0
1025 GOTO 1055
1030 PRINT AT 11,19;"5"
1031 PRINT TAB 19;"."
1034 PRINT AT 16,17;"OU "
1035 LET U=0
1036 IF N=1 OR O=1 OR P=1 OR Q=1
 THEN LET U=1
1037 GOTO 1055
1038 PRINT AT 11,17;"4"
1039■PRINT AT 12,17;"."
1040 IF M(L,6)=0 THEN GOTO 1088
1043 PRINT AT 11,19;"■"
1046 PRINT AT 12,19;"*"
1049 PRINT AT 16,17;"ET "
1050 LET U=0
1051 IF (N=1 AND O=1) OR (N=1 AN
D P=1) OR (N=1 AND Q=1) OR (O=1
AND P=1) OR (O=1 AND Q=1) OR (P=
1 AND Q=1) THEN LET U=1
1055 IF U=0 THEN GOTO 1075
1060 PRINT AT 14,21;"*****"
1065 PRINT AT 17,21;"***"
1070 GOTO 1100
1075 PRINT AT 14,21;"....."
1080 PRINT AT 17,21;"..."
1085 GOTO 1100
1088 PRINT AT 11,19;"5"
1090 PRINT AT 12,19;"."
1093 LET W=1
1095 PRINT AT 16,17;"FIN"
1100■IF M(L,7)=0 THEN GOTO 1135
1105 PRINT AT 11,24;"■"
1110 FOR X=12 TO 16
1111 PRINT AT X,24;"*"
1112 NEXT X
1115 IF U=0 THEN GOTO 1148
1120 LET S=1
1125 PRINT AT 17,25;"***"
1126 PRINT AT 17,28;S
1127 PRINT AT 19,28;"*"
1130 PRINT AT 20,13;"************
****"
1134 GOTO 1200
1135 PRINT AT 11,24;"6"
1140 FOR X=12 TO 16
1141 IF X=14 AND U=1 THEN GOTO 1
143
1142 PRINT AT X,24;"."
1143 NEXT X
1144 PRINT AT 17,25;"..."
1145 GOTO 1200
1148 LET S=0
1150 PRINT AT 17,25;"..."
1155 PRINT AT 17,28;S
1160 PRINT AT 19,28;"."
1163 PRINT AT 20,13;"............
...."
1200 IF M(L,8)=0 THEN GOTO 1235
1205 PRINT AT 11,26;"■"
1210 PRINT AT 12,26;"*"
1211 PRINT AT 13,26;"*"
1215 IF U=0 THEN GOTO 1245
1220 LET V=1
1223 PRINT AT 14,27;"***"
1224 PRINT AT 14,30;V
1225 FOR X=16 TO 21
1226 PRINT AT X,30;"*"
1227 NEXT X
1228 FOR Y=30 TO 10 STEP -1
1229 PRINT AT 21,Y;"*"
1230 NEXT Y
1231 FOR X=21 TO 16 STEP -1
1232 PRINT AT X,10;"*"
1233 NEXT X
1234■GOTO 1293
1235 PRINT AT 11,26;"7"
1240 PRINT AT 12,26;"."
1241 PRINT AT 13,26;"."
1242 PRINT AT 14,27;"..."
1243 GOTO 1293
1245 LET V=0
1250 PRINT AT 14,27;"..."
1252 PRINT AT 14,30;V
1255 FOR X=16 TO 21
1256 PRINT AT X,30;"."
1257 NEXT X
1260 FOR Y=30 TO 10 STEP -1
1261 PRINT AT 21,Y;"."
1262 NEXT Y
1265 FOR X=21 TO 16 STEP -1
1266 PRINT AT X,10;"."
1267 NEXT X
1293 IF W=1 THEN GOTO 1350
1295 IF K$="R" THEN GOTO 1320
1300 PRINT AT 20,1;"TAPEZ T "
1310■IF INKEY$="T" THEN GOTO 131
8
1315 GOTO 1310
1318 GOSUB 9000
1320 GOSUB 6000
1325 GOTO 475
1350 PRINT AT 20,1;"TAPEZ Z"
1355 IF INKEY$="Z" THEN GOTO 136
3
1360 GOTO 1355
1363 GOSUB 9000
1365 GOSUB 5000
1370 GOTO 405
5000 LET T=0
5005 FOR N=0 TO 4
5010 PRINT AT 6,N;"*"
5015 PRINT AT 6,(N+1);"*"
5020 PRINT AT 6,N;"."
5025 NEXT N
5030 PRINT AT 6,5;"■"
5033 GOSUB 8000
5035 GOSUB 7000
5040 IF K=0 THEN GOTO 5050
5045 PRINT AT K,17;".."
5050 RETURN
6000 LET T=T+1
6003 IF T>7 THEN GOTO 120
6005 FOR N=0 TO 4
6010 PRINT AT 4,N;"*"
6015 PRINT AT 4,(N+1);"*"
6020 PRINT AT 4,N;"."
6025 NEXT N
6030 PRINT AT 4,5;"■"
6033 GOSUB 8000
6035 IF T=1 OR T=3 OR T=5 OR T=7
 THEN LET A$="1"
6040 IF T=2 OR T=3 OR T=6 OR T=7
 THEN LET B$="1"
6045 IF T=4 OR T=5 OR T=6 OR T=7
 THEN LET C$="1"
6050 GOSUB 7000
6055 LET K=T+1
6060 RETURN
7000 PRINT AT 5,7;T
7005 PRINT AT 4,9;A$
7010 PRINT TAB 9;B$
7015 PRINT TAB 9;C$
7020 IF A$="1" THEN PRINT AT 4,1
0;"****"
7025 IF A$="0" THEN PRINT AT 4,1
0;"...."
7030 IF B$="1" THEN PRINT AT 5,1
0;"****"
7035 IF B$="0" THEN PRINT AT 5,1
0;"...."
7040 IF C$="1" THEN PRINT AT 6,1
0;"****"
7045 IF C$="0" THEN PRINT AT 6,1
0;"...."
7050 PRINT AT (T+1),17;"**"
7055 IF T>0 THEN PRINT AT T,17;"
.."
7060 RETURN
8000 LET A$="0"
8005 LET B$="0"
8010 LET C$="0"
8015 RETURN
9000 PRINT AT 19,1;"            "
9005 PRINT AT 20,1;"            "
9010 RETURN
```

GOSUB 5000 : Exécution RAZ
GOSUB 6000 : Exécution TOP
GOSUB 7000 : Affichage T dans le compteur
Affichage du nombre binaire à la sortie du compteur
Affichage des fils adresse
Affichage de laigne-mémoire
Affichage de A§, B§, C§
GOSUB 8000 : Remise à zéro de A§, B§, C§
GOSUB 9000 : Effacement des lignes Dialogue 19 et 20.

* Ecriture des instructions en mémoire

1ère instruction :
PAS E1 = R2 10001110 8Eh
Faire une remise à zéro RAZ. Touche Z
Introduire par les touches Ø et 1 le mot ci-dessus (les bits 1 apparaissent en vidéo inversée).
Envoyer l'ordre E. Touche E
Le mot s'inscrit en mémoire, les bits 1 sont représentés par E

2ème instruction :
E2 ET E4 = R2 Ø1Ø1Ø11Ø 56h
Envoyer un TOP Touche T
Introduire le 2ème mot Touches Ø et 1
Envoyer l'ordre E Touche E
Procéder de même pour les instructions suivantes.

3ème instruction : PAS E2 = R1 Ø1ØØ11Ø1 4Dh
4ème instruction : E1 ET E3 = R1 1Ø1ØØ1Ø1 A5h
5ème instruction : E3 OU E4 = R1 ØØ111ØØ1 39h
6ème instruction : Arrêt ØØØØØØØØ ØØh

Faire une RAZ (touche Z) pour réinitialiser la mémoire.

* Lecture des instructions et exécution

Le programme (ensemble des instructions d'un traitement) ainsi introduit, il reste à entrer les données E1 et E2 (Ø ou 1) par les touches Ø et 1.
Le traitement peut être effectué suivant deux modes :

Le mode PAS A PAS pour mieux suivre le déroulement (touche P) : vous envoyez vous-mêmes les Tops T et les ordres K.
. Envoyer l'ordre lecture "Touche L"
La 1ère instruction est lue en mémoire et, par le BUS, commande le traitement.
. Quand le résultat est affiché, envoyer le 1er Top "Touche T".
La 2ème instruction est lue et est exécutée.
Et ainsi de suite...
La dernière instruction stoppe le traitement.

Le mode RUN : le traitement s'opère en continu "Touche R"

Vous pouvez recommencer le traitement sur de nouvelles données. Pour exécuter un autre programme, faire BREAK, RUN et NEWLINE.

Claude Remy

initiation

Un ordinateur comment ça marche ?

L'ordinateur a envahi tous les compartiments de la vie moderne. D'abord réservé aux grosses sociétés, il s'est petit à petit, introduit partout au fur et à mesure que son prix diminuait et que sa technique s'affinait.

En lui ajoutant le son et la couleur, il est devenu micro-ordinateur familial pour lequel on assiste actuellement à une véritable explosion.
Le merveilleux ordinateur qu'est le ZX81, a été le détonateur de cette explosion. Sa réussite a donné naissance à un tas d'autres micro-ordinateurs. Depuis, Sinclair a sorti le ZX Spectrum dont les performances, du point de vue qualité/prix, n'étaient pas imaginables il y a quelques années à peine.

Les ordinateurs, et les micro-ordinateurs, sont habituellement comparés entre eux suivant la puissance de leurs jeux d'instructions, le nombre de routines dont ils disposent, les modes d'adressage qu'ils comportent, leurs étendues mémoire, leurs possibilités graphiques et leurs vitesses d'exécution. Autant de choses qui traduisent un fonctionnement interne complexe que le non-initié a du mal à comprendre. L'ambition de cette brève introduction est de clarifier un peu ces problèmes en montrant la simplicité de certains principes de base, valables depuis les origines.

DE BLAISE PASCAL A CLIVE SINCLAIR

Les principes de base d'une petite machine à calculer ont été développées par Pascal, qui avait réalisé une additionneuse entièrement mécanique. Les principes d'une calculatrice moderne ont été mis au point par Babbage vers les années 1830. Mais c'est Von Neumann, en 1945, qui a introduit le véritable ordinateur en appliquant deux concepts nouveaux fondamentaux :

1 - Le programme enregistré : le résultat est une calculatrice programmable dont le programme est figé dans une mémoire.
2 - Le rupture de séquence : en cours d'exécution du programme, la machine peut prendre une décision si telle condition est réalisée et cela, grâce à un branchement conditionnel ou à un saut qui rompt la séquence normale du programme.

Les machines basées sur les conceptions de Von Neumann sont toujours employées actuellement bien que des conceptions nouvelles soient appliquées aux gros ordinateurs.

On distingue plusieurs générations dans l'évolution des ordinateurs :

1ère génération : vers 1950 l'ordinateur à tubes électroniques qui remplissait une pièce.
2ème génération : vers 1962 l'ordinateur à transistors qui remplissait une console.
3ème génération : vers 1970 l'ordinateur à circuits intégrés qui remplit un tiroir.
4ème génération : vers 1975 l'ordinateur à circuits L.S.I. qui remplit une carte.

En plus de la miniaturisation évidente, des progrès ont été accomplis parallèlement en fiabilité, en prix et en vitesse d'exécution. L'ordinateur a suivi l'évolution des circuits intégrés. Ceux-ci, au départ,

dans la série 7400, avaient quelques portes intégrées sur une puce de silicium. Depuis, on en a intégré des milliers. C'est d'ailleurs la dimension et non la complexité de la puce qui détermine son prix.
Les différentes étapes ont été, grosso modo :

- S.S.I. (Small Scale Integration) : intégration à petite échelle comportant quelques dizaines de portes par puce,
- M.S.I. (Medium Scale Integration) : intégration à moyenne échelle comportant quelques centaines de portes par puce,
- L.S.I. (Large Scale Integration) : intégration à grande échelle comportant quelques milliers de portes par puce.

On parle maintenant d'une 5ème génération d'ordinateurs avec le V.L.S.I. (Very Large Scale Integration). Mais tout ne peut pas se miniaturiser. Il y a l'interface humain : l'homme doit pouvoir manipuler l'ordinateur en appuyant sur des touches, en tournant des clés, des combinateurs, etc...

Les dimensions des ZX81 et Spectrum sont à leur minimum possible : le clavier doit rester accessible à son utilisateur. Par contre, ce qu'il y a en dessous du clavier évoluera encore. Nous aurons des circuits plus denses, plus performants. Nous aurons plus de mémoires, plus de routines, plusieurs langages de programmation, plus d'interfaces pour imprimantes et autres périphériques, etc...

DANS LA BOITE

La constitution d'un ordinateur, basée sur la logique et le calcul binaire, reste fondamentalement la même, qu'il s'agisse d'un gros, d'un mini ou d'un micro-ordinateur. La distinction entre ces classes d'ordinateurs réside dans la dimension des mots traités. Les gros ordinateurs traitent des mots de 32 bits, les minis, des mots de 16 bits et les micros, de 8 bits. On parle, bien sûr, de micro-ordinateurs de 16 bits. Ce ne sont que des minis qui se miniaturisent. L'évolution de la technologie les rend compétitifs.

La figure 1 montre la constitution d'un ordinateur. L'unité 1, qu'on appelle processeur central ou unité centrale de traitement ou microprocesseur, est le coeur du système. Dans les ZX, on emploie le microprocesseur Z80 fabriqué par Zilog. C'est ce processeur qui exécute les opérations, qui range les résultats, qui décode les instructions à réaliser et qui établit la séquence du déroulement de l'exécution. Pour toutes ces fonctions, le processeur central emploie des signaux de commande C, pour commander la mémoire centrale et l'unité d'entrées/sorties. En réponse à ces signaux, les unités 2 et 3 envoient et reçoivent des informations que nous rangeons sous le vocable général d'opérandes O.

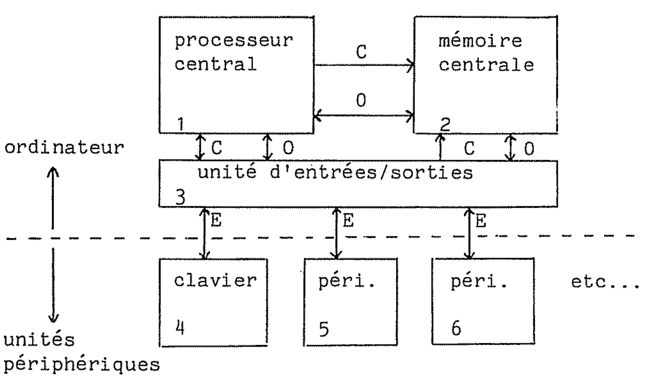


figure 1 - constitution d'un ordinateur.

L'unité 2, qui est la mémoire centrale, est constituée d'un ensemble de cellules. Dans les ZX, la mémoire centrale est subdivisée en MEM (ou ROM) et en MEV (ou RAM). Chaque cellule contient une information ; nous en distinguons deux sortes :

1 - Les instructions rangées séquentiellement en mémoire et dont l'ensemble constitue un programme. Le processeur central en commande l'extraction afin de les décoder (MEM et MEV).
2 - Les opérandes sur lesquelles vont s'effectuer les différentes opérations décodées par le processeur (MEV uniquement).

Nous voyons que la mémoire a un rôle passif mais très important. Elle subit les commandes C aussi bien de la part du processeur que de l'unité 3. Ces commandes sont de deux sortes :

1 - Les commandes de lecture, auquel cas la mémoire fournit l'information d'une cellule au demandeur, (MEM et MEV).
2 - Les commandes d'écriture et la mémoire enregistre dans une cellule l'information venant des unités 1 ou 3 (MEV uniquement).

Tel quel, l'ordinateur fonctionne parfaitement. Cependant tout ce qu'il fait reste inconnu. Pour compléter notre ordinateur, il lui faut une 3ème unité, dite d'entrées/sorties qui interface les deux unités précédentes au monde extérieur. Elle établit des communications entre l'ordinateur et les unités extérieures qu'on appelle unités périphériques à l'aide de signaux d'échange E. Dans les ZX, cette fonction est remplie par le circuit intégré spécialisé et spécifique à Sinclair. Dans les ZX les périphériques prévus, donc interfacés, sont : un clavier, un enregistreur, l'imprimante Sinclair, un modulateur pour T.V. et un petit H.P. (Spectrum). La présence d'un connecteur arrière permet le raccordement à tout autre périphérique : mémoire supplémentaire, imprimante commerciale, manette de jeux, etc... mais chaque fois avec un circuite d'interfaçage adéquat.

La configuration d'un système de traitement d'informations est déterminée par l'utilisateur. En fonction du problème à résoudre, l'utilisateur établit la liste et les caractéristiques des unités dont il a besoin et la manière de les faire travailler. L'utilisateur doit donc aborder plusieurs techniques pour atteindre son but.
Le matériel (ou hardware) qui réunit la technologie, les circuits logiques, etc... est tout ce qui est matériel, visible, palpable. C'est le domaine de l'électronicien.
Le logiciel (ou software) qui réunit le langage de programmation, le système d'exploitation, etc... est tout ce qui est immatériel, invisible, spirituel. C'est le domaine du programmeur.

Signalons à ce sujet, deux particularités des ZX qui rendent son utilisation tellement agréable. Le fait d'introduire les mots-clés à l'aide d'une seule touche rend la programmation plus rapide.

Deuxièmement, le contrôleur de syntaxe qui signale les erreurs d'une ligne éditée avant son introduction dans un programme. (Ne pas confondre avec les compte-rendus d'erreur que tous les micro-ordinateurs possèdent avec plus ou moins de codes et de texte).
Ces deux particularités réduisent fortement les erreurs de programmation et le temps qu'on consacre à leurs recherches.

Marcel Henrot

LOGIGRAPH

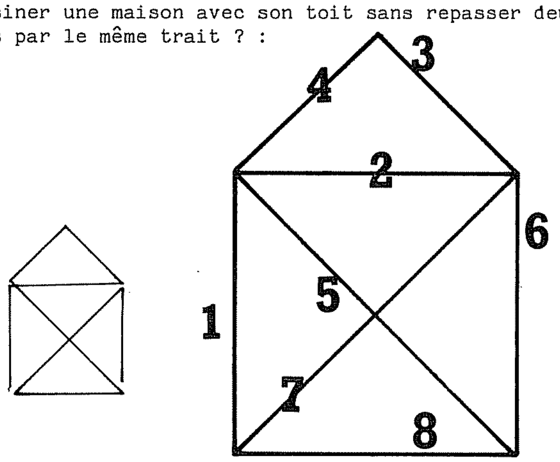
Yves Leclerc, le célèbre auteur canadien d'ouvrages sur l'informatique individuelle, qualifie le Langage LOGO, inventé par Seymour Papert, de "convivial" ; et ce terme est assez bien trouvé pour ce langage naturel, d'accès facile et spécialement adapté à l'initiation scolaire.

Les listes de LOGO sont très lisibles puisque chaque instruction symbolise nettement une action concrète orientée vers le graphisme : le programmeur s'identifie à une "petite tortue" qui commence à marcher au milieu de l'écran et qui change de direction, avance de tant de points, rechange de direction etc, et accomplit ainsi un parcours qui définit un graphisme plus ou moins élaboré.

Les interpréteurs LOGO sont très consommateurs de mémoire puisque chaque instruction représente une action complète ; il en résulte que ce langage, et c'est un paradoxe qu'on peut relever, est peu répandu sur les ordinateurs domestiques alors qu'il est tout indiqué pour l'apprentissage informatique des enfants. Il n'existe pas, à notre connaissance, de cassette développée pour Sinclair, mais ceci ne tardera sans doute pas. Pour vous préparer à cet évènement, B. Clergeot vous propose un programme d'initiation aux principes du LOGO sous la forme d'un logiciel de simulation avec un dictionnaire déjà étoffé.

Mise en oeuvre du programme

Un fois la liste entrée et l'ordre RUN lancé, l'ordinateur demande de rentrer des chaînes de caractère : il faut rentrer successivement les différentes instructions qui composent le petit programme de démonstration (n° 1) dessinant un rectangle : attention ne pas inscrire les numéros de ligne, le Spectrum numérote automatiquement. Quand toute la liste est entrée, introduire "l" et vérifier qu'il n'y a pas de faute dans la liste ; puis "e" et voyez le rectangle se dessiner. Voilà vous avez compris le principe... Etudiez attentivement les deux autres programmes également reproduits pour vous donner des idées, et, en choisissant vos instructions dans le "dictionnaire - logigraph", avec l'aide des indications données dans "l'éditeur" pour la correction de vos listes, essayez maintenant de programmer ce petit casse-tête classique : comment dessiner une maison avec son toit sans repasser deux fois par le même trait ? :



LOGIGRAPHE
----

* Capacité : 100 lignes
* Variables : 10 indicées x : v (x)
* Label : 20 indicées x : Label x
* Longueur d'une ligne : Maximum 29 caractères

EDITEUR

Le programme se présente sous forme de lignes numérotées.
Pour entrer une instruction on n'entre par le n° de ligne : on les entre à la suite, puis on retire où ajoute des lignes.

Instructions : Mode direct

l : liste les 22 premières lignes.
l x : liste à partir de x.
i : fait une place entre deux lignes en décalant la ligne du dernier listing vers le bas.
d : supprime la ligne du dernier listing et remonte d'une ligne celles du dessous.
e : exécute le programme.
Sauve : sauve le programme sur cassette.
Oublier : perd le programme.
Variable v (x) : affiche la variable V(x).

DICTIONNAIRE "LOGIGRAPH"

Instructions : mode interprété par le BASIC.

V(x)=n : la variable V(x) prend la valeur n.

Direction n : la direction en cours devient n.

Position nnn,nnn : le point de début du graphisme est est le PLOT nnn,nnn.

Efface : efface l'écran.

Papier x : le fond devient de la couleur de x.

Encre x : l'encre est de la couleur de x.

Afficher en nn,nn : affiche ce qui est écrit à la ligne suivante aux coordonnées d'écran nn,nn.
Exemple : 10 AFFicher en 10,10
11 Bonjour je suis le Spectrum

Label x : positionne un label où il sera fait un saut.

Son : fait un son bref et aigue.

Avance x : avance dans la direction en cours de x

Avance V(x) : avance de V(x)

Gauche x : la direction en cours vire à gauche de x degrés

Gauche V(x) : la direction en cours vire de V(x) degrés

Droite x : la direction en cours vire à droite de x degrés

Droite V(x) : la direction en cours vire à droite de V(x) degrés

Fin : fin du programme

Va au label x : saut inconditionnel au label x

Si V(x)=0 va au label x : saut inconditionnel : si la variable V d'indice x est égale à 0

Liste x : fait en programme la liste des 22 lignes suivant x, x inclus.

Calcul de v(x) = ... : la variable V(x) prend la valeur du résultat de ce qui suit le égal.

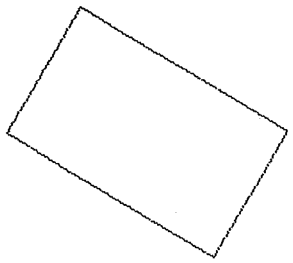
Presser le space arrête le programme.

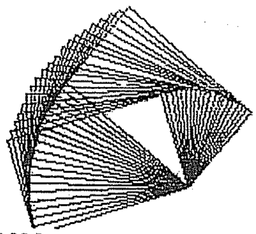
Dans les calculs vous pouvez utiliser la variable.

h a s a r d qui est comprise entre 0 et 1.

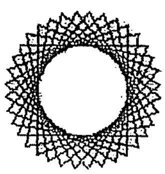
En cas d'arrêt du programme : faire GOTO edit.

```
1  effacer
2  direction 30
3  position 170,10
4  avance 90
5  gauche 90
6  avance 150
7  gauche 90
8  avance 90
9  gauche 90
10 avance 150
11 fin
```





```
1  effacer
2  position 170,050
3  v(1)=40
4  v(2)=20
5  label 1
6  direction v(1)
7  avance 60
8  gauche 90
9  avance 100
10 gauche 90
11 avance 60
12 gauche 90
13 avance 100
14 calcul de v(1)=v(1)-3
15 calcul de v(2)=v(2)-1
16 si v(2)=0 va au label 2
17 va au label 1
18 label 2
19 fin
```

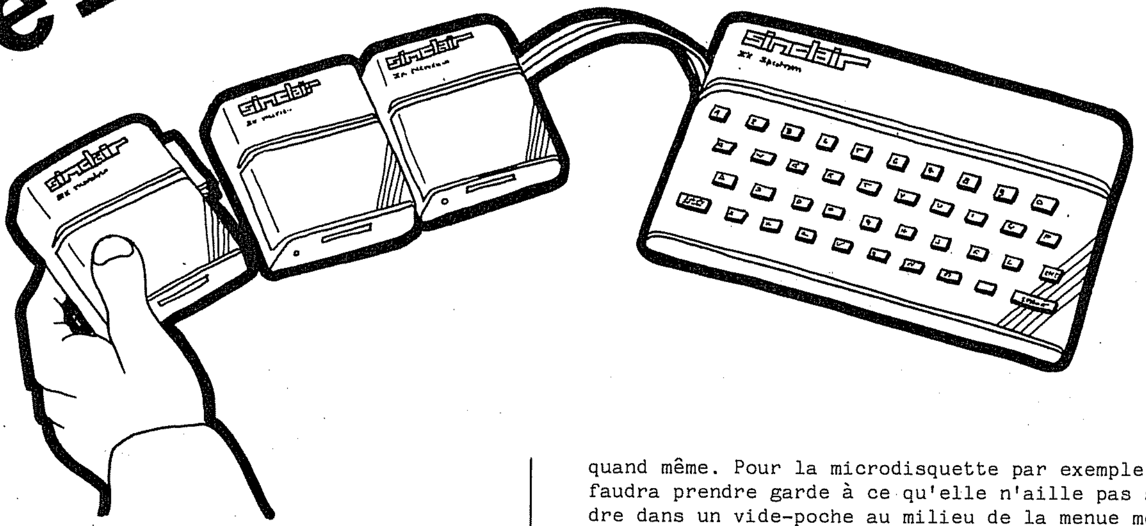


```
1  papier 0
2  encre 7
3  efface
4  direction 0
5  position 140,050
6  afficher en 00,05
7  programme en logigraphe
8  v(1)=36
9  label 2
10 avance 80
11 gauche 110
12 calcul de v(1)=v(1)-1
13 si v(1)=0 va au label 1
14 va au label 2
15 label 1
16 son
17 papier 7
18 encre 0
19 fin
```

```
   5 PRINT AT 0,5;"©CLERGEOT Ber
trand                   1983"
   6 PRINT AT 5,10;"LOGIGRAPHE"
  10 REM ******logigraphe******
  20 LET D=0
  30 LET K=360/(2*PI)
  40 DIM I$(100,29)
  45 DIM l(20)
  50 DIM v(10)
  60 LET edit=8000
  80 GO TO 5000
  90 REM *****interpreteur*****
 100 LET co=1
 101 PLOT 128,87
 105 LET c$=I$(co)
 108 LET hasard=RND
 110 IF c$(1 TO 6)="avance" THEN
 GO SUB 1000
 120 IF c$(1 TO 7)="gauche " OR
c$(1 TO 7)="droite " THEN GO SUB
 1100
 130 IF c$(1 TO 3)="fin" THEN GO
 TO 5000
 140 IF c$(1 TO 6)="efface" THEN
 CLS
 150 IF c$(1 TO 9)="direction" T
HEN LET D=VAL c$(11 TO )
 160 IF c$(1 TO 8)="position" TH
EN PLOT VAL c$(10 TO 12),VAL c$(
14 TO 16)
 170 IF c$(1 TO 11)="va au label
" THEN LET co=l(VAL c$(13 TO ))
 180 IF c$(1 TO 2)="v(" THEN LET
 v(VAL c$(3))=VAL c$(6 TO )
 190 IF c$(1 TO 5)="si v(" THEN
GO SUB 4000
 200 IF c$(1 TO 9)="calcul de" T
HEN LET v(VAL c$(13))=VAL c$(16
TO )
 210 IF c$(1 TO 6)="papier" THEN
 PAPER VAL c$(8 TO )
 220 IF c$(1 TO 5)="encre" THEN
INK VAL c$(7 TO )
 230 IF c$(1 TO 5)="liste" THEN
LET p=VAL c$(7 TO ): GO SUB 6110
 240 IF c$(1 TO 8)="afficher" TH
EN PRINT AT VAL c$(13 TO 14),VAL
 c$(16 TO 17);I$(co+1): LET co=c
o+1
 250 IF c$(1 TO 3)="son" THEN BE
EP .05,30
 700 IF INKEY$=" " THEN PRINT AT
 21,0;"arret en ";co: GO TO 5000
 800 LET co=co+1
 999 GO TO 105
1000 IF c$(8)="v" THEN DRAW SIN
(D/K)*v(VAL C$(10)),COS (D/K)*v(
VAL C$(10)): RETURN
1010 DRAW SIN (D/K)*VAL C$(8 TO
),COS (D/K)*VAL C$(8 TO ): RETUR
N
1100 IF c$(1)="d" THEN LET d=d+(
VAL c$(8 TO )): RETURN
1110 IF c$(1)="g" THEN LET d=d-(
VAL c$(8 TO )): RETURN
1120 GO SUB 9000
1130 RETURN
3000 PRINT "preparation des labe
ls"
3002 FOR n=1 TO 100
3005 IF I$(n)(1 TO 5)="label" TH
EN LET l(VAL I$(n)(7 TO ))=n
3010 NEXT n
3020 GO TO 100
4000 IF v(VAL c$(6))=0 THEN LET
co=l(VAL c$(23 TO ))
4010 RETURN
5000 REM ******editeur*****
5005 LET P=1
5020 INPUT C$: IF c$="" THEN GO
TO 5020
5021 LET c$=c$+" "
5022 LET T=0
5023 IF c$="sauve " THEN SAVE "p
rog. logi" DATA I$(): GO TO 5000
5024 IF c$="lire " THEN LOAD "pr
og. logi" DATA I$(): GO TO 5000
5025 GO TO 6000
5030 LET I$(P)=C$
5032 IF c$="oublier " THEN RUN
5035 PRINT P;TAB 3;I$(p)
5040 LET P=P+1: IF P=101 THEN LE
T P=100
5900 GO TO 5020
6000 IF c$(1 TO 2)="l " THEN GO
SUB 6100
6010 IF c$="i " THEN GO SUB 6200
6020 IF c$="d " THEN GO SUB 6300
6040 IF c$="e " THEN GO TO 3000
6042 IF c$(1 TO 3)="var" THEN GO
 SUB 6400
6045 IF T=1 THEN GO TO 5020
6050 GO TO 5030
6100 CLS : IF LEN c$=2 THEN LET
c$=c$+"1"
6101 LET p=VAL c$(3 TO )
6102 IF p>100 THEN GO TO 6125
6110 FOR n=p TO p+21
6111 IF n>100 THEN GO TO 6125
6115 IF i$(n)(1)=" " THEN GO TO
6125
6120 PRINT n;TAB 3;I$(n): NEXT n
6125 LET T=1
6128 PRINT AT 0,0;
6130 RETURN
6200 LET T=1
6205 FOR n=99 TO p STEP -1
6210 LET I$(n+1)=I$(n)
6230 NEXT n
6240 LET I$(p)=">"
6250 LET c$="l "+STR$ p: GO SUB
6100
6260 RETURN
6300 FOR n=p TO 99
6305 LET I$(n)=I$(n+1): NEXT n
6306 LET I$(100)=" "
6310 LET c$="l "+STR$ p: GO SUB
6100
6320 RETURN
6400 CLS
6410 PRINT "v(";c$(12);")=";v(VA
L c$(12)): LET T=1: RETURN
8000 CLS : PRINT "Instruction: "
;c$: PRINT "erreur en ligne ";co
: PAUSE 0: LET c$="l "+STR$ co:
GO TO 5021
9000 IF d>360 THEN LET d=d-360
9005 IF d<0 THEN LET d=360+d
9010 RETURN
9100 FOR n=0 TO 2*PI STEP .1
```

B. Clergeot

Le microdrive, enfin !



Certains d'entre vous vont dire : "ORDI-5 nous parle du Microdrive alors qu'on en est encore à attendre le Spectrum". Pourtant nous ne résistons pas au plaisir de vous en parler ; d'abord parce que ce n'est pas si prématuré : les Anglais commencent à être livrés d'une manière réguliere, ensuite parce qu'il multiplie à tel point les possibilités du Spectrum que certains pourraient bien attendre qu'il soit là pour trouver un grand frère à leur ZX81 avant de tourner trop vite vers une certaine concurrence (suivez mon regard...) ; enfin et surtout parce que nous avons trop envie de vous décrire ce bijou que la rédaction a eu le privilège de tenir dans les mains grâce à la gentillesse du directeur général de la firme Sinclair lui-même.

IL TIENT DANS LE CREUX DE LA MAIN

Un vrai bijou donc, c'est l'impression que fait le microdrive quand on le sort de sa boîte. On l'avait bien vu en photo et on le savait petit pas à ce point quand même. Pour la microdisquette par exemple, il faudra prendre garde à ce qu'elle n'aille pas se perdre dans un vide-poche au milieu de la menue monnaie : elle est vraiment microscopique. Mettre un timbre dans une boîte d'allumettes de cuisine, ce sont à peu près les dimensions. Soyons plus précis, le drive lui-même mesure 90 mm sur 85 mm avec une épaisseur de 40 mm. La microdisquette, que nous appellerons la cartouche, est un petit rectangle de 45 syr 35 mm avec 7 mm d'épaisseur.
Le Microdrive n'a pas seulement la taille du bijou il en a la beauté : un design qui rappelle le Spectrum lui-même, une belle couleur noire avec le label en arc-en-ciel sur le coin inférieur droit, et tous les éléments en sont ajustés avec une grande précision : une fois la cartouche insérée dans la fente prévue à cet effet il n'y a pas le moindre jeu.

Le Microdrive en revanche s'éloigne d'une troisième caractéristique propre aux bijoux : le prix ; il est en effet actuellement vendu en Angleterre pour la somme de 49 livres soit 600 F : étant donné qu'il tient les promesses techniques (annoncées voilà quelques temps déjà !), on a l'impression que c'est comme si on pouvait acheter une voiture de sport pour le prix d'une mobylette. Nous avons eu très envie d'ouvrir le ventre du microdrive pour voir comment on pouvait faire aussi petit et puis nous nous sommes tout de même raisonnés : ce n'est pas le moment de tout casser, avant de faire de la chirurgie, attendons de pouvoir se procurer un microdrive de rechange dans la première boutique spécialisée comme une simple cassette vierge : la leçon d'anatomie ce sera pour un prochain numéro d'ORDI-5.

100 FOIS PLUS RAPIDE !

Venons en à l'essentiel, la capacité et la vitesse : les microdrive sont des cartouches à bande sans fin qui reprennent une technique déjà utilisée dans les "wafertape" mais nettement miniaturisée : la bande est plus étroite que celle d'une "microcassette" classique ; il y a environ 5 mètres de ruban qui passent à la vitesse de 80 centimètres à la seconde devant la tête de lecture : il s'ensuit qu'il suffit environ d'une dizaine de secondes pour lire la totalité de la bande sur un tour.
Le microdrive est prévu pour 100K (en fait un peu moins : 85 K étant un minimum toujours garanti).

Un programme de 48K est lu en moins de 5 secondes si la bande est positionnée en début de fichier : la lecture de programmes pré-enregistrés se fait avec une vitesse multipliée par un facteur 100 ! Un bruit de feulement assez fort et assez soutenu en dit long sur la vitesse de déplacement.

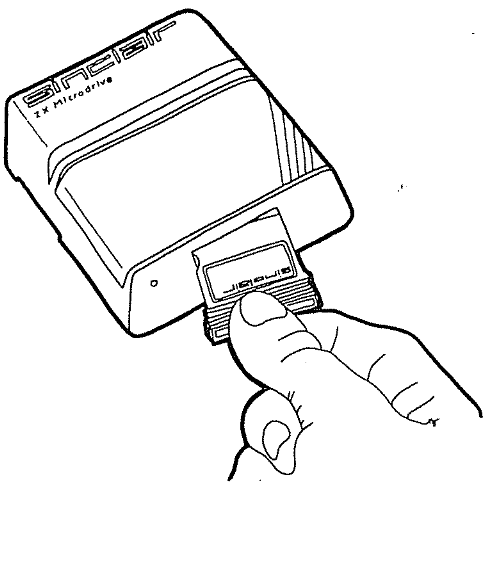
LA DEUXIEME MERVEILLE

Tel quel, le drive ne peut pas être utilisé en conjonction directe avec le Spectrum, il est nécessaire d'ajouter une interface, dite "interface 1" qui constitue elle aussi une merveille de rapport possibilité/prix puisque pour 350 FF on dispose d'un système non seulement capable de piloter une bande sans fin mais aussi de permettre les développements de télécommunication et le couplage avec des imprimantes rapides : il il s'agit de l'interface RS 232.

Cette merveille est la partie modeste du dispositif ; modeste parce qu'elle vient s'accoler à l'arrière du Spectrum et en dessous (ce qui en passant donne à l'appareil une position plus ergonomique), mais modeste également car l'essentiel de la publicité n'a pas porté sur cet élément qui constitue pourtant une bonne part de la prouesse technique de Sinclair : proposer une interface RS232 à ce prix. Il est clair que nos amis Anglo-américains peuvent avec un minimum d'argent se doter d'un ordinateur absolument complet et sans équivalent dans cette catégorie.

BOND DANS LE DOMAINE PRO

Ces capacités considérablement accrues permettent d'envisager sans sourire de véritables applications professionnelles pour le Spectrum - nous entendons par véritables celles qui ne demandent pas à l'utilisateur un trop gros effort de manipulation, puisqu'aussi bien le ZX81 est capable de supporter des applications professionnelles importantes quand on est capable de surmonter la rusticité des opérations de sauvegarde ou chargement. -



Un domaine dans lequel l'amélioration de la vitesse de chargement peut avoir une influence décisive est celui des jeux : la frustration que l'on éprouve à voir les marbrures de l'écran pendant quelques minutes avant de pouvoir jouer vont disparaître ; on peut aussi penser à des échanges de jeux et de logiciel entre clubs en glissant simplement dans une enveloppe la microcartouche de quelques grammes : et nous parlons du courrier "avion" puisque la cartouche ne pèse que 8 grammes !

Entre un Spectrum ordinaire et un Spectrum muni de son microdrive et de son interface il n'y a pas qu'une simple différence de dégré, il y a une véritable opposition de nature : on a affaire à un autre ordinateur comme on s'en rend tout de suite compte avec l'utilisation d'une batterie tout à fait nouvelle de commandes.

Quiconque n'a pas utilisé les disquettes des gros ordinateurs de table mettra même du temps à se rendre compte de toutes les possiblités qu'il y a à utiliser en accès immédiat 100 K de mémoire en masse, en ouvrant et en fermant les fichiers, en appelant en désordre différents programmes de travail etc...

Après un certain temps d'utilisation on s'aperçoit que le véritable drame consiste à ne disposer que d'une seule microcartouche livrée d'origine avec l'appareil : il s'agit encore pour l'instant d'une denrée rare ; on voudrait pouvoir tout mettre dans ces 100 K, et l'on arrive vite à saturation (mais oui !). Vivement la production de masse ! Signalons une difficulté à ce niveau : la transposition de programmes Spectrum enregistrés sur cassete, qui font usage de routines en langage machine, nécessitent une adaptation, car le branchement de microdrives modifie sensiblement les zones de mémoire RAM ; cet inconvénient n'est pas rencontré avec les programmes tout BASIC.

EXPLICATION DES MYSTERIEUSES INSTRUCTIONS DU CLAVIER

CAT : L'ordre CAT affiche sur l'écran en 10 secondes les titres de tous les fichiers (programmes ou données) contenues sur la cartouche à concurrence de 50 titres.

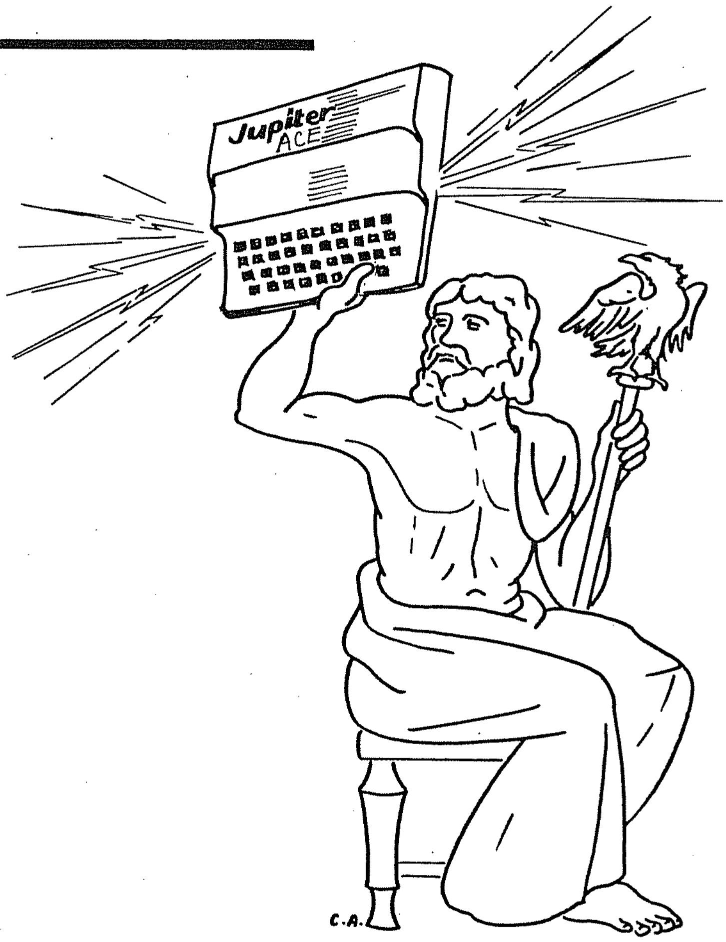
FORMAT : Prépare la microcartouche en lui donnant un nom.
ERASE : Efface sur la cartouche les programmes ou données que l'on veut écraser.
OPEN ≠ et CLOSE ≠ ouvrent et ferment les fichiers : un fichier fermé ne peut être écrasé par erreur.
MOVE : Permet de transférer les informations d'un canal à un autre (écran, clavier, microdrive, sont des canaux).

Le paramètrage de toutes ces commandes, comme des commandes classiques LOAD, SAVE et VERIFY, peut être assez complexe étant donné la multiplication des possibilités par exemple.

SAVE *"m";1"ORDI-5" LINE 1 signifie : sauvegarder sur la première cartouche le programme ORDI-5 de telle sorte que, chargé, il s'exécute à la première ligne.

Henri Davignet

Modulons notre FORTH



Il y a deux "environnements" très différents pour les programmeurs : l'interprétation et la compilation qui conduisent à des méthodes de programmation très différentes.

La première méthode est née avec les micro-ordinateurs programmables en BASIC. Ce dernier interprété, permet une programmation aisée, facilite les modifications et autorise la programmation "fouilli" (il y invite même d'ailleurs). Son principal avantage est qu'il est conversationnel lors de l'écriture des programmes.

La deuxième méthode date des ordinateurs où les programmes étaient entrés par des cartes perforées, puis compilés et enfin exécutés. La programmation ainsi faite est souvent fastidieuse car longue mais a l'avantage de fournir des programmes plus "propres", lisibles et où l'on sait toujours exactement ce qui se passe (enfin presque toujours !).

FORTH présente cet avantage à peu près unique de combiner l'interprétation pour l'élaboration des programmes et la compilation pour leur exécution parce que la programmation progresse d'une façon modulaire : on ne peut pas travailler une nouvelle instruction (mot) sans avoir compilé les instructions précédentes;ceci permet des corrections rapides du programme, des essais de petits morceaux (grâce à sa modularité) et une programmation très fortement structurée (pour qui veut s'en donner la peine).

Le programme peut donc, lorsqu'il a été dégrossi sur le papier, être entré directement dans le micro-ordinateur et essayé sans plus attendre.

Malheureusement, et c'est là le principal défaut de FORTH, à l'état brut, sans commentaire, un programme est illisible. Il faut donc toujours fournir l'effort de documenter chaque module sous peine d'être incapable de s'en servir huit jours après.

Nous allons illustrer cette idée de structuration modulaire en programmant un petit jeu simple et connu :
- L'ordinateur génère un nombre aléatoire que le joueur doit trouver en proposant successivement des nombres. Des indications de supériorité ou d'infériorité lui sont données pour chaque tentative.

Lorsque la réponse est trouvée, l'ordinateur donne le nombre de tentatives et termine.

LE NOMBRE SECRET

L'analyse s'effectuera en deux étapes :
- d'abord l'analyse des modules du programme principal
- ensuite l'analyse des différentes instructions de chaque module.

Le tout sans effort logique démesuré.

PROGRAMME

- Initialisation
  Nombre de propositions = Ø
- Aléatoire (appel du programme de tirage au sort du chiffre)
- Proposition
- Si Egaux alors Terminé
  sinon Supérieur ?
- Incrémenter le nombre de propositions (Incrémenter c'est ajouter 1)
- Attendre (pour éviter un effacement immédiat du résultat de la proposition)
- Reprendre à proposition

Fin programme ;

Aléatoire

(Ce sous programme est appelé par son nom dans le programme principal)
- Tirer un nombre aléatoirement ;
- Le mettre sur la pile

Fin d'aléatoire ;

Proposition

- Ecrire 'Entrez votre proposition :'
- Ecrire ' (nombre entre 1 et 50) '
- Attendre un nombre
- Si nombre > 50 alors écrire 'Erreur' et charger le nombre en Ø
  sinon mettre le nombre sur la pile

Fin proposition ;

Egaux

Si les 2 nombres sur la pile sont égaux
  alors écrire 'C'est gagné !'
  écrire nombre proposition 'Coups'
  mettre oui sur la pile (c'est pour dire au programme principal que c'est terminé)
  sinon mettre non sur la pile
Fin égaux ;

Supérieur ?

Si le nombre aléatoire est > à la proposition
  alors écrire 'Trop grand !'
  sinon écrire 'Trop petit !'
Fin supérieur ? ;

Attendre

Faire 800 fois rien (une boucle qui ne fait rien que perdre du temps)
Fin attendre ;

LE PROGRAMME

```
: RND (Construire ce mot comme dans le manuel du
       Jupiter Ace page 85)
: CHIFFRE (mot de création d'une entrée numérique)
  QUERY (Demande un nombre)
  NUMBER (Transfère sur la pile ce nombre et son
          adresse)
  DROP   (Laisse tomber l'adresse)
 ;
Ø VARIABLE NBRPROPOSITION (Crée la variable de stoc-)
                          (kage du nombre de Proposi-)
                          (tions du Joueur. Initiali-)
                          (se la variable à Ø)

 (Génération du nombre à trouver en aléatoire)
: ALEATOIRE (le : met en mode compile et on crée le )
            (nom du nouveau module.                 )
  5Ø RND 1+ (le nombre sera entre 1 et 5Ø           )
;           (fin de ce module.                      )

: PROPOSITION (Nouveau module de nom PROPOSITION    )
    CLS       (Efface l'écran                       )
    ." Entrez votre proposition "
    CR        (Passe à la ligne suivante            )
    ." Un nombre entre 1 et 5Ø puis retour chariot:")
              (on vient d'écrire une question sur   )
              (l'écran                              )
    CHIFFRE   (on attend l'entrée du joueur         )
 ;
(Test de l'égalité entre la proposition et le nombre)
(à trouver                                          )
: EGAUX     (Nom du nouveau module                  )
    OVER    (copie du nombre à trouver sur la pile  )
    OVER    (copie de la proposition sur la pile    )
    =       (test de l'égalité                      )
    IF  CR ." GAGNE !!!! "(on écrit que le joueur a )
                          (gagné                    )
        CR ." En "        (on écrit le nombre de coups)
        NBRPROPOSITION @.(nécessaire pour gagner    )
        ."coups"
        1     (on laisse 1 sur la pile car ils sont )
              (égaux                                )
  ELSE        (sinon on laisse Ø sur la pile pour   )
        Ø     (dire qu'ils ne sont pas égaux        )
  THEN        (on sort du module                    )
 ;
(Test si la proposition est supérieure ou inférieure)
: SUPERIEUR?
    OVER    (copie de nombre à trouver sur la pile)
    >       (la proposition est supérieure ?      )
    IF
      CR   ."TROP GRAND !"    ( oui               )
    ELSE
      CR   ."TROP PETIT !"    (non                )
    THEN
 ;                            (on sort du module  )


(Module de temporisation)
: ATTENDRE   (Nom du module                        )
    8ØØØ     (De 8ØØØ on décrémente                )
    Ø        (à Ø                                  )
    DO       (On fait une boucle qui ne fait rien  )
    LOOP     (On sort du module                    )
 ;
(Module principal du jeu                           )
: TROUVE               (Nom d'appel du jeu         )
    ALEATOIRE          (On tire le nombre à        )
                       (trouver                    )
    BEGIN              (Début des recherches       )
      PROPOSITION      (Le joueur tente un chiffre )
      EGAUX            (A-t-il trouvé le bon       )
                       (chiffre                    )
      IF   DROP DROP EXIT (Oui on vide la pile et on)
                       (sort                       )
        ELSE SUPERIEUR? (Non trop grand ou trop    )
                       (petit ?                    )
        THEN
      NBRPROPOSITION @ (On met le nombre d'essais)
      1+               (sur la pile et on ajoute )
      NBRPROPOSITION ! (1 avant de le restocker  )
      ATTENDRE         (On attend avant de recom-)
      Ø UNTIL          (mencer à cause de l'effa-)
 ;                     (cement.                  )
```

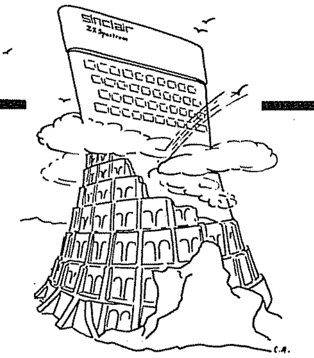
CE QUI SE PASSE DANS LA PILE A CHAQUE INSTRUCTION

ALEATOIRE	: VIDE	→	Nombre à trouver
BEGIN	: Nombre à trouver	→	Nombre à trouver
PROPOSITION	: Nombre à trouver	→	Proposition Nombre à trouver
EGAUX	: Proposition Nombre à trouver	→	Ø ou 1 (si pas égaux (ou si égaux Proposition Nombre à trouver
IF	: Ø ou 1 Proposition Nombre à trouver	→	Proposition Nombre à trouver

Si le sommet de la pile était 1, la pile est vidée (DROP,DROP) et le programme est arrêté (EXIT).

SUPERIEUR?	: Proposition Nombre à trouver	→	Nombre à trouver
NBRPROPOSITION	: Nombre à trouver	→	Adresse de la variable Nombre à trouver
@	: Adresse de la variable Nombre à trouver	→	Nombre de proposition Nombre à trouver
1+	: Nombre de proposition Nombre à trouver	→	Nombre de proposition +1 Nombre à trouver
NBRPROPOSITION	:		
!	: Adresse de la variable Nombre de proposition +1 Nombre à trouver	→	Nombre à trouver
ATTENDRE	: Pas d'effet sur la pile		
Ø UNTIL	: Pas d'effet sur la pile. On met Ø sur la pile avant d'exécuter UNTIL car on veut être sûr de recommencer à chaque fois et on reprend juste après BEGIN.		

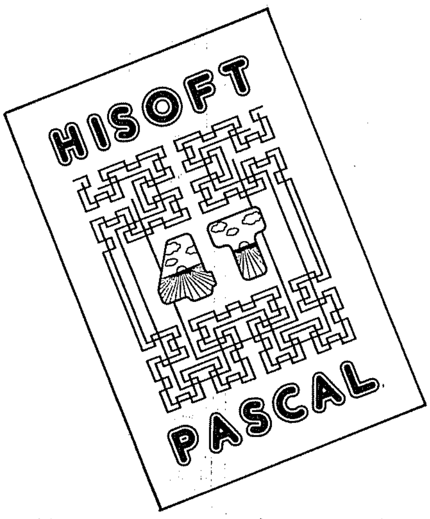
Régis Senegou



PASCAL sur ZX

La cassette Pascal pour Spectrum 48 K de chez Hisoft est une réalistion de qualité professionnelle dont le succès est tel en Angleterre qu'aucun concurrent ne s'est encore risqué à proposer une cassette du même type et que c'est plutôt celle-ci qui est transposée pour d'autres matériels que le Spectrum.

Chacun sait qu'un compilateur Pascal tient en mémoire une place considérable : l'étude de ce langage ne pouvait donc être menée valablement que sur des ordinateurs moyens tels l'Apple. On peut maintenant véritablement apprendre le Pascal sur un ordinateur domestique car le compilateur et le programme d'interfaçage n'occupent qu'une vingtaine de K de mémoire. Nous disons bien <u>apprendre</u> le Pascal et non simplement programmer en Pascal ; en effet, quoi qu'en dise la documentation technique qui prétend que cette cassette est faite pour les gens qui savent déjà programmer dans ce langage, nous estimons, nous, qu'elle est aussi parfaitement adaptée à l'étude progressive de celui-ci étant donné la manière particulièrement explicite dont l'éditeur signale les fautes de syntaxe.



Cette cassette a donc une portée assez générale et, à condition d'avoir fait au préalable l'acquisition du supplément mémoire nécessaire, il s'agit d'un "package" qui en donne énormément pour pas cher : on peut sans prendre de risque mettre quiconque à défi de commencer à programmer en Pascal pour moins de 300 F.

Comme la chose se produit très souvent pour les cassettes de grande qualité, le manuel n'est pas, mais pas du tout, à la hauteur du programme. Ce qui vérifie une fois de plus un phénomène souvent rencontré : que d'excellents programmeurs peuvent être de piètres pédagogues et de mauvais écrivains : chacun son métier, évidemment ! il en résulte de très nombreux tâtonnements avant de savoir quelle touche provoque telle réaction ou quel est l'effet de telle fonction à tel stade : pour parodier les Anglais nous dirions que la documentation est "extremy confusing". Nous avons mis des heures à comprendre comment on pouvait déclencher l'option imprimante du compilateur pour vous lister le petit programme de résolution d'équation du second degré ci-dessous : il faut faire (*!P*) dans un commentaire, cela ressemble à un phylactère d'injure dans une bande dessinée...

Tout ce qui pourrait transformer le manuel en un ouvrage praticable est absent : il n'y a pas d'organigramme général pour illustrer la hiérarchie des différentes commandes ou le moyen simple de passer de l'un des trois états (BASIC, Editeur, Compilateur) à l'un des deux autres ; il n'y a pas un seul exemple correctement développé pour faire la découverte pas à pas des différentes possibilités de la cassette ; il n'y a pas d'index permettant de s'y retrouver dans le livre et enfin il n'y a aucun plan rationnel ; zéro donc pour le manuel.

```
             10 PROGRAM EQ;
AB5A    20 VAR A,B,C:REAL;
AB63    30 D,R,I:REAL;
AB63    40 BEGIN
AB8C    50 READ (A,B,C);
AB8A    60 R:=-B/(2*A);
ABB7    70 D:=B*B-4*A*C;
ABF8    80 I:=SQRT(ABS(D))/(2*A);
AC22    90 IF D>0
AC33   100 THEN WRITELN('Les raci
nes sont: ',R+I,' et ',R-I)
ACA4   110 ELSE IF D=0
ACBB   120 THEN WRITELN('Racine d
ouble: ',R)
ACEF   130 ELSE WRITELN('Racines
complexes: ',R,'+i',I,'et ',R,'-
i',I)
AD63   140 END.
End Address: AD68
```

```
             10 PROGRAM AB;
AB2E    20 VAR A,B:INTEGER;
AB37    30 BEGIN
AB40    40 READ (B);
AB46    50 FOR A:=1 TO B DO
AB64    60 BEGIN
AB67    70 WRITE('*');
AB6C    80 END;
AB70    90 END.
End Address: AB72
```

```
10 INPUT A
20 FOR B=1 TO A
30 PRINT "*";
40 NEXT B
```

UN LOGICIEL D'EXCELLENCE !

Heureusement si l'on s'accroche deux bonnes heures en substituant sa logique à celle des rédacteurs on découvre l'une après l'autre les extraordinaires qualités de ce logiciel et l'on prend vite beaucoup de plaisir à utiliser les différentes fonctions d'édition qui ont un air très professionnel (il en est ainsi possible de commander l'incrémentation automatique des instructions comme le renumérotage ou l'effacement partiel et sélectif du programme, etc.) ; le plaisir s'accroît naturellement lorsque l'on commande la compilation qui s'effectue sous nos yeux ligne par ligne à une vitesse très appréciable; enfin l'exécution des programmes objet est d'une grande rapidité comparé à l'équivalent en BASIC sans atteindre naturellement à la rapidité de l'Assembleur "cousu main". Ce gain de rapidité est d'un facteur 5 à 10 ainsi que nous l'avons testé sur le petit programme numéro 2 qui consiste à couvrir l'écran d'astérisque (1 seconde contre 5).

Nous aurons certainement l'occasion de vous reparler de ce logiciel et nous faisons même un "appel au peuple" pour quelques beaux programmes réalisés de cette manière à insérer dans les colonnes d'ORDI-5.

X. L. B.

Réalisez un port de sortie 8 bits

Avec une dizaine de circuit intégrés vous pouvez réaliser une extension qui permet au ZX de commander des éléments extérieurs, de petits automatismes par exemple.

PRINCIPE DE FONCTIONNEMENT

On dispose de 8 sorties prenant chacune un état 1 ou 0, ces sorties sont utilisables individuellement ou sous la forme d'un octet : sous cette dernière forme la valeur de la sortie sera comprise entre 0 et 255 (00 à FF en hexadécimal).
Ce circuit n'occupe qu'une seule adresse pour sortir l'octet de donnée ; cette adresse est sélectionnée par straps entre 0 et 65535 :
On compare les 16 bits d'adresse (A0...A15) avec la valeur sélectionnée par 16 straps (qui forment l'adresse binaire désirée). S'il y a égalité de valeur, on mémorise l'octet correspondant à cette adresse et présent sur le bus des données.
Donc le circuit réagit simplement à la présence de l'adresse voulue sans se préoccuper des signaux de contrôle du microprocesseur tels que $\overline{MREQ}$, $\overline{RD}$ ou $\overline{WR}$.

DESCRIPTION DU SCHEMA

Les bus d'adresses et des données sont tout d'abord envoyés sur des buffers (IC1...IC4) qui permettent de ne pas trop charger les sorties du microprocesseur et de pouvoir utiliser ces informations pour d'autres extensions sans perturber le bon fonctionnement du ZX.
Depuis la sortie des buffers, les 16 bits d'adresse vont sur un comparateur 16 bits (sur les entrées A).
Ce comparateur est réalisé avec 4 (IC6...IC9) comparateurs 4 bits (74LS85) montés en cascade.

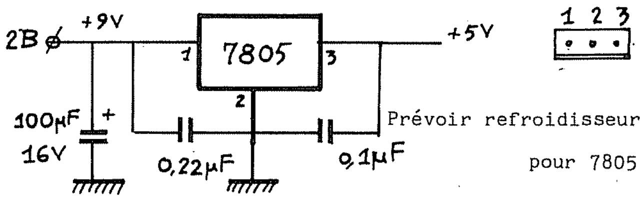
On sélectionne l'adresse désirée en positionnant les straps branchés sur les entrées B du comparateur.
La sortie 6 de IC9 sera au niveau haut si, et seulement si, toutes les comparaisons sont vraies.

On transforme la sortie du comparateur en impulsion par l'intermédiaire de R1 et C1. Cette impulsion sert à charger les 8 bascules (74LS75 x 2) servant de mémoire pour les données. Les straps S1 et S2 sont prévus pour d'autres extensions et permettent si besoin est de bloquer les buffers.

RECOMMANDATIONS POUR LE MONTAGE

Pour la construction du montage on suivra les recommandations suivantes :

- Prévoir un régulateur +5V pour l'alimentation de ce circuit (consommation ≃ 200 mA).



face métallique vue côté connexions du 7805

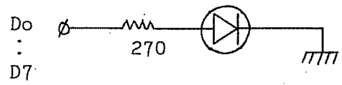
- Mettre un condensateur de 0,1μF en parallèle sur l'alimentation des CI tous les 2 circuits.
- Ramener les 8 fils de sortie ainsi que +9V, +5V et 0V sur un connecteur (ou une barrette de dominos) pour l'utilisation.
- A la place des 16 straps on peut installer :
  . 2 blocs d'interrupteurs en boîtier DIL 16 broches
  . 4 roues codeuses hexadécimales.

LISTE DES COMPOSANTS

74LS75 x 2
74LS85 x 4
74LS367 x 4
7805 x 1 refroidisseur
Résistance 100Ω x 1
Condensateur : 100μF/16V x 1
1μF/16V x 1
0,1μF x 6
0,22μF x 1

UTILISATION DE L'EXTENSION

Pour se familiariser avec le montage on peut brancher 8 LED en sortie :



Puis on sélectionne une adresse d'utilisation.
Dans le manuel du ZX, il est indiqué que l'adresse 16417(d) est inutilisée ; nous l'utiliserons donc pour les essais.
Pour passer d'une adresse décimale à son équivalent en binaire 16 bits on peut procéder comme suit :

16417(d) = 256 x 64 + 33

Page 181 à 187 du manuel du ZX, on trouve la correspondance entre code hexa et code décimal, qui donne :

64(d) = 40(h) et 33(d) = 21(h)
donc 16417(d) = 4021(h)

D'autre part, l'équivalent binaire des chiffres hexa (0...F) est :

0	0000	8	1000
1	0001	9	1001
2	0010	A	1010
3	0011	B	1011
4	0100	C	1100
5	0101	D	1101
6	0110	E	1110
7	0111	F	1111

4021(h) s'écrira donc : 0100 0000 0010 0001
(A15) (Ao)

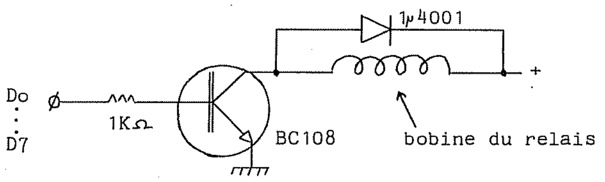
et l'on postionnera les straps en conséquence en sachant qu'un strap mis représente un Ø.
Entrer le programme suivant :

```
10 FOR A = Ø TO 255
20 POKE 16417,A
30 NEXT A
et RUN
```

On verra les LED s'allumer en fonction de la valeur binaire de A.

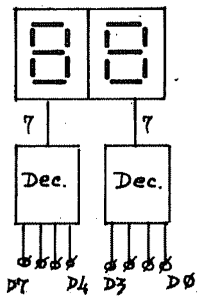
APPLICATIONS

1) - Commande de relais :



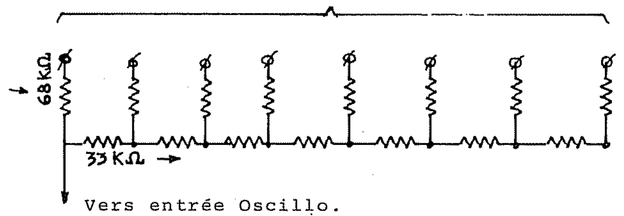
2) - Lecture d'une case mémoire contenant une variable.
Dans la mise au point d'un programme en langage machine, on se sert souvent de cases mémoires pour stocker une variable de calcul.
Exemple : 2 D (26000),A

On aimerait voir les différentes valeurs de la variable pendant l'exécution du programme, ce qui est souvent difficile à obtenir facilement. L'extension rendra de grands services, car positionnée sur cette adresse (dans l'exemple : 260 00(d)), elle nous indiquera le nouveau contenu à chaque changement.
Il est possible, à la place des 8 LEDs en sortie qui ne sont pas commodes à lire, de monter 2 décodeurs hexadécimaux pour 2 afficheurs 7 segments. Chaque décodeur sera commandé par 4 bits.
Exemple :



3) - Commande d'un autre appareil (exemple : terminal vidéo) en appliquant l'octet de sortie à l'entrée parallèle. Il suffira pour la commande de faire succéder les différentes valeurs.
Peut permettre de calculer en mode FAST et d'afficher sur un autre écran les résultats et sur un format qui n'est pas limité à 32 colonnes.
Il faudra bien évidemment utiliser des routines en langage machine.

4) - Utilisation de l'informatique pour l'électronique.



A l'aide du petit montage ci-contre et avec un oscilloscope - lorsque l'on est équipé - on peut visualiser des signaux de toutes les formes.

Programme d'application :

```
10 FOR A = Ø TO 255
20 POKE 16417,A
30 NEXT A
40 FOR A = 255 TO Ø STEP -1
50 POKE 16417,A
60 NEXT A
70 GOTO 10
```

mode FAST et RUN permet de produire des signaux triangulaires de fréquence très basse. La vitesse est limitée par le BASIC.

On peut augmenter la vitesse en utilisant une routine en langage machine.

Taper le programme suivant :

```
1 REM XXXXXXXXXX... (32 fois un caractère)
10 FOR A = 16514 TO 16544
20 INPUT B
30 POKE A,B
40 NEXT A
50 RAND USR 16514
60 GOTO 50
```

Rentrer les nombres suivants lors de l'exécution du programme :

6,255,120,50,33,64,16,250,201,STOP

Détail du programme en assembleur :

```
LD B,255
LD A,B
LD(16417),A
DJNZ-6
RET
```

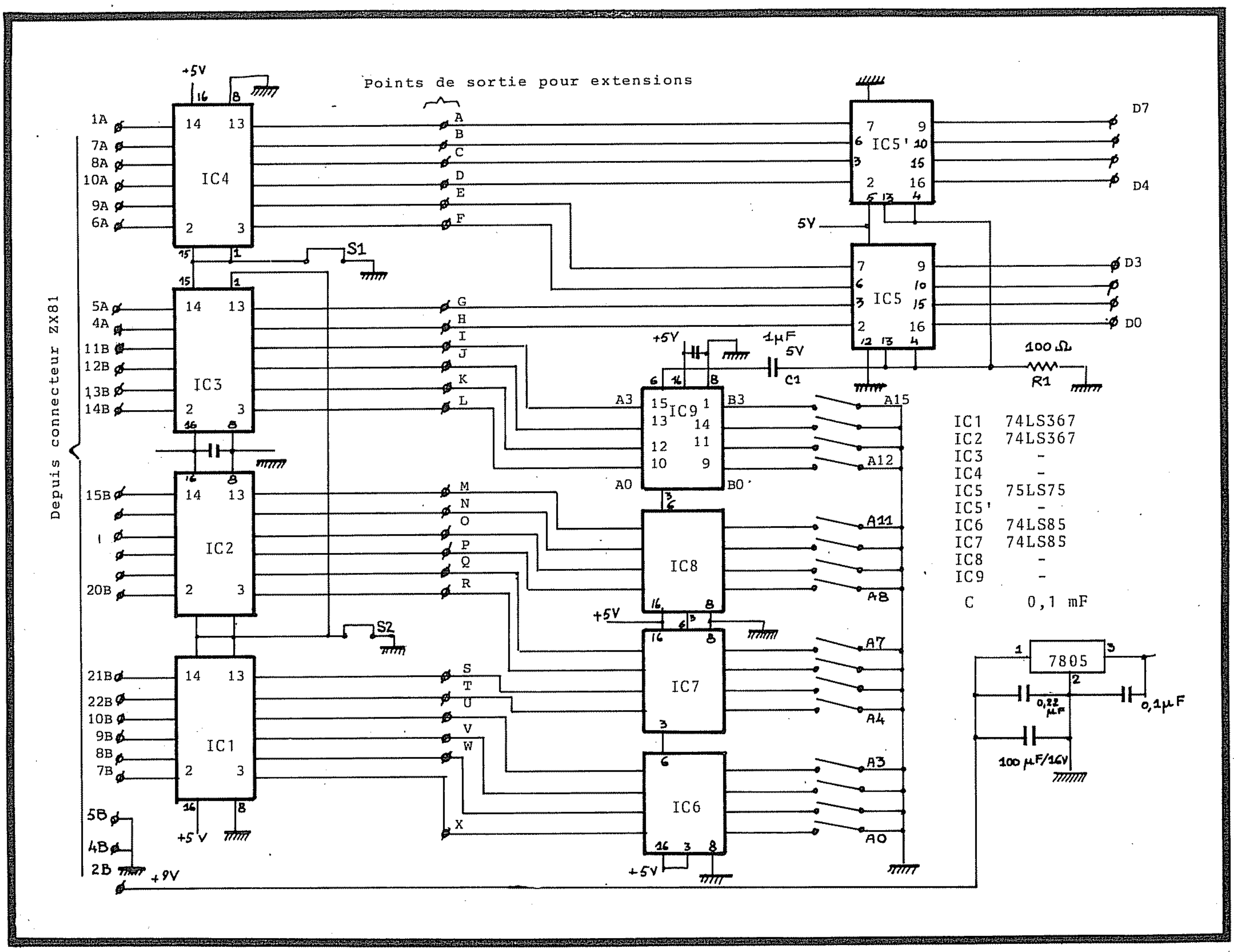
en mode FAST on fait RUN 50, et on observe sur l'oscilloscope des signaux en dents de scie (fréquence ≃ 400 Hz).
Pour produire des signaux triangulaires, rentrer les nombres suivants lors de l'exécution du programme :

6,255,120,50,33,64,16,250,50,33,64,60,254,255,56,248, 205,(70),15,56,235,201,STOP

Détail du programme en assembleur :

```
LD B,255
LD A,B
LD(16417),A
DJNZ-6
LD(16417),A
INC A
CP 255
JR C,-8
CALL $OF46
JRC,-21
RET
```

On supprime la ligne 60 du programme et en mode FAST, on fait RUN 50. On observe des signaux triangulaires de fréquence 180 Hz environ.
Pour stopper le programme, il faut appuyer sur la touche BREAK. Il se peut que sur certains "vieux ZX 81" on ne puisse pas interrompre le programme, il faut alors remplacer dans la liste des nombres à entrer le nombre entouré par 67.




```
  0 REM GESTION SORTIES
 10 DIM A$(8)
 15 FOR A=1 TO 8
 16 LET A$(A)="0"
 17 NEXT A
 20 CLS
 30 PRINT "INTRODUIRE LE NO DE
LA SORTIE"
 40 INPUT A
 45 IF A>8 OR A<1 THEN GOTO 40
 50 PRINT ,,"TAPER ""0"" OU ""1
"""
 60 IF INKEY$="" OR INKEY$<>"0"
AND INKEY$<>"1" THEN GOTO 60
 70 LET A$(A)=INKEY$
 80 LET F=0
 90 LET D=F
100 FOR C=8 TO 1 STEP -1
110 LET D=D+VAL A$(C)*(2**F)
120 LET F=F+1
130 NEXT C
140 POKE 16417,D
150 PRINT ,,"ETAT DES SORTIES"
160 PRINT ,," NO ";
170 FOR X=2 TO 9
160 PRINT AT 6,X*2;X-1;AT 7,X*2
;A$(X-1)
190 NEXT X
200 IF INKEY$="" THEN GOTO 200
210 GOTO 20
```

Programme 1 : Permet de changer une seule sortie à la fois en spécifiant son numéro et la valeur à mettre Ø ou 1.

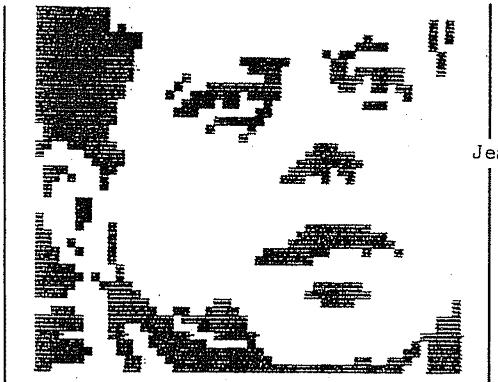
```
  1 REM GESTION SORTIES
  2 REM EX: ADRESSE 16417
 10 LET A$=""
 20 PRINT ,,"ENTRER ""0"" OU ""
1""","POUR CHAQUE SORTIE"
 30 FOR E=7 TO 0 STEP -1
 40 PRINT AT 10,7-E;E
 50 IF INKEY$="" THEN GOTO 50
 55 IF INKEY$<>"0" AND INKEY$<>
"1" THEN GOTO 50
 60 LET A$=A$+INKEY$
 70 PRINT AT 11,0;A$
 80 NEXT E
 90 LET B=8
100 LET F=0
110 LET D=F
120 FOR C=B TO 1 STEP -1
130 LET D=D+VAL A$(C)*(2**F)
140 LET F=F+1
150 NEXT C
160 POKE 16417,D
170 IF INKEY$="" THEN GOTO 170
180 CLS
190 GOTO 10
```

Programme 2 : Permet de changer plusieurs sorties en même temps mais oblige à réentrer la valeur de chaque sortie (Ø ou 1).

G. Pedroli

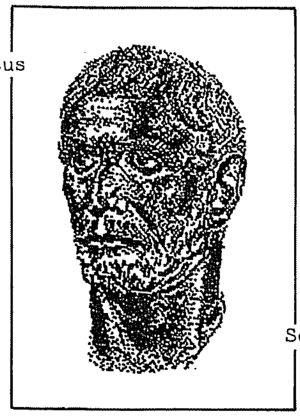
Ecran, mon bel écran

Si vous avez la chance de posséder ce bijou qu'est l'imprimante Sinclair, vous pouvez nous envoyer les plus beaux "écrans" que vous avez réalisés, quel qu'en soit le genre.



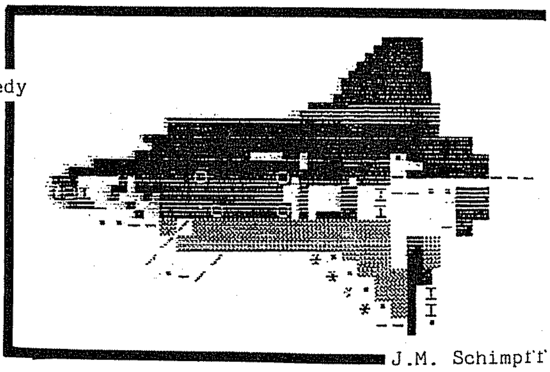
De Kennedy (Dejà 20 ans !)

Jean-Luc Papillon



Brutus

Scherer



Navette spatiale à Cap Kennedy

J.M. Schimpff

Côté court

Peinture abstraite

ZX81 1K

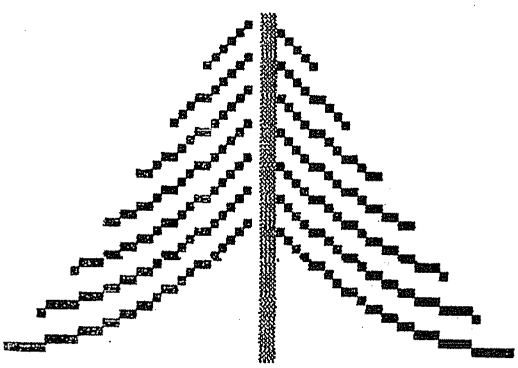
```
10 REM PEINTURE ABSTRAITE
20 REM     EVOLUTIVE
30 LET I=INT (RND*2)
40 LET A=INT (RND*11)
50 IF I=1 THEN LET A=A+128
60 LET X=INT (RND*22)
70 LET Y=INT (RND*32)
80 PRINT AT X,Y;CHR$ A
90 GOTO 20
```

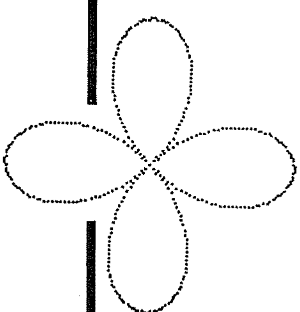
Jean Casenave

```
 5 FOR J=1 TO 28 STEP 4
10 FOR I=J TO 30
20 PLOT I,I*I/60+J
30 PLOT 63-I,J+I*I/60
40 NEXT I
50 NEXT J
60 FOR I=1 TO 21
70 PRINT AT I,16;"▓"
90 NEXT I
```

Freddie Blin

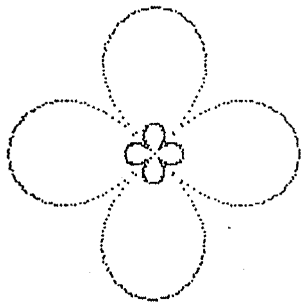
Ah! mon beau sapin





Roses de Noël

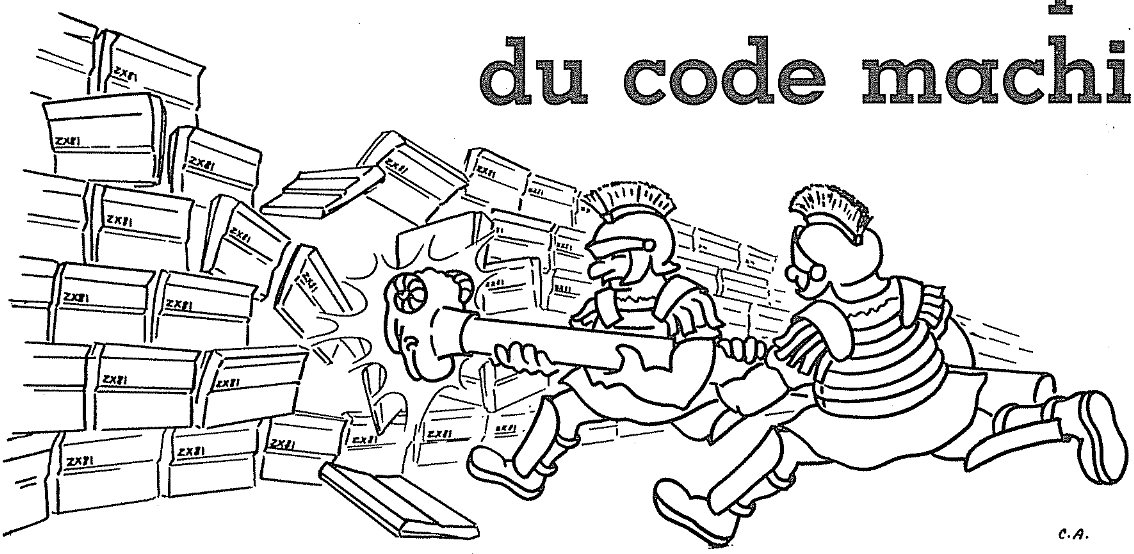
```
 1 REM © MORANDINI R. & ORDI-5
 5 LET C=50: LET D=50
10 FOR B=0 TO 360
20 LET A=(PI*B)/180
30 LET X=C*(SQR ABS (2*COS (2*
A))*COS A
40 LET Y=D*(SQR ABS (2*COS (2*
A))*SIN A
50 LET F=X+127: LET H=Y+88
60 PLOT F,H
70 NEXT B
75 IF C=10 THEN STOP
80 LET C=10: LET D=10
90 GO TO 10
```



Morandini

```
 1 REM © MORANDINI R. & ORDI-5
10 FOR B=0 TO 360
20 LET A=(PI*B)/180
30 LET X=50*SQR 2*COS (2*A)*CO
S A
40 LET Y=50*SQR 2*COS (2*A)*SI
N A
50 LET F=X+127: LET H=Y+88
60 PLOT F,H
70 NEXT B
```

Devenez un champion du code machine



Suivez pas à pas la construction d'un jeu de casse-brique particulièrement performant avec balles à effet et accroissement progressif de la vitesse d'exécution. Les erreurs d'introduction sont difficiles à envisager car codes machines et mnémoniques assembleur font l'objet de deux listages parallèles. Il est tout à fait possible d'utiliser un assembleur ZX81 sur cassette ; après le jeu de Bernard Lacoste vous trouverez une petite note sur les principales cassettes d'assembleur actuellement distribuées en France.

UN CASSE-BRIQUE A TOUT CASSER

Ce jeu est composé de :

1) Une partie en BASIC de l'instruction 15 à l'instruction 240 pour :
- dessiner le terrain
- centraliser les différents paramètres (score, vitesse, etc...)
- afficher les scores et gérer les parties supplémentaires.

2) Une partie en C.M. abritée dans les REM 1 à 5, pour le jeu lui-même ;

3) Les instructions 1000 à 1240 forment un programme de chargement des C.M, dérivé de celui de Marcel Henrot, pour charger des instructions en hexadécimal de un à quatre octets ;

4) Le programme BASIC de 2000 à la fin permet d'examiner le contenu des REM.

La partie BASIC

Elle s'analyse sans difficulté à partir du listing joint (instructions 15 à 240).
Les variables utilisées sont :

MS : meilleur score obtenu
SC : score cumulé (partie en cours)
ST : score du tableau qui vient d'être terminé
V : vitesse du jeu (augmente quand V diminue)
DC : vitesse horizontale de la balle (± nombre de colonnes)
DL : vitesse verticale de la balle (± 1 ligne x 33)
DCN : vitesse horizontale de la balle normée (SGN(DC))

```
  15▒LET MS=0
  17 LET SC=0
  20 LET V=1000
  21 POKE 16856,INT (V/256)
  22 POKE 16855,V-256*INT (V/256
)
  23 POKE 17071,INT (V/256)
  24 POKE 17070,V-256*INT (V/256
)
  35 CLS
  40 PRINT " ▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒
▒▒▒▒▒▒▒▒▒▒"
  50 FOR I=1 TO 3
  55 PRINT "▌
          ▐"
  60 NEXT I
  70 FOR I=4 TO 12
  75 PRINT " ▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒
▒▒▒▒▒▒▒▒▒▒"
  80 NEXT I
  85 FOR I=13 TO 21
  90 PRINT "▌
          ▐"
  95 NEXT I
 100 LET DC=INT (4*RND+.5)-2
 105 LET DCN=SGN DC
 110 LET DL=-33
 115 POKE 16516,DC
 116 POKE 16517,0
 117 POKE 16518,DL
 118 POKE 16519,0
 119 POKE 16522,DCN
 120 POKE 16523,0
 121 POKE 16528,5
 122 POKE 16529,0
 123 POKE 16530,0
 150 LET ST=USR 16801
 160 LET SC=SC+ST
 165 PRINT AT 15,5;"SCORE JEU ="
;ST;" BRIQUES"
 170 PRINT TAB 5;"SCORE PARTIE =
";SC;" BRIQUES"
 175 IF MS<SC THEN LET MS=SC
 180 PRINT TAB 5;"MEILLEUR SCORE
=";MS;" BRIQUES"
 185 IF ST>200 THEN GOTO 210
 190 PRINT ,,"LA PARTIE EST TERM
INEE.","APPUYEZ SUR UNE TOUCHE"
 195 PAUSE 40000
 200 GOTO 17
 210 PRINT ,,"LA PARTIE CONTINUE
..."
 211 LET V=V-100
 212 IF V<100 THEN LET V=100
 215 PAUSE 100
 220 GOTO 21
 230 SAVE "C▒"
 240 GOTO 0
1000 PRINT "ADRESSE DE DEBUT: ";
1005 INPUT AI
1010 PRINT AI
1015 LET NI=AI
1020 IF PEEK 16442<3 THEN SCROLL
1025▒INPUT A$
1027 IF A$="S" THEN STOP
1030 IF LEN A$<2 THEN GOTO 1025
1035 IF LEN A$>2 THEN GOTO 1060
1040 POKE NI,(16*CODE A$+CODE A$
(2)-476)
1045 PRINT NI;" ";A$;
1050 LET DI=1
1055 GOTO 1180
```

```
1060 IF LEN A$<4 THEN GOTO 1025
1065 IF LEN A$>4 THEN GOTO 1100
1070 POKE NI,(16*CODE A$+CODE A$
(2)-476)
1075 POKE NI+1,(16*CODE A$(3)+CO
DE A$(4)-476)
1080 PRINT NI;" ";A$(1 TO 2);" "
;A$(3 TO 4);
1085 LET DI=2
1090 GOTO 1180
1100 IF LEN A$<6 THEN GOTO 1025
1105 IF LEN A$>6 THEN GOTO 1140
1110 POKE NI,(16*CODE A$+CODE A$
(2)-476)
1115 POKE NI+1,(16*CODE A$(3)+CO
DE A$(4)-476)
1120 POKE NI+2,(16*CODE A$(5)+CO
DE A$(6)-476)
1125 PRINT NI;" ";A$(1 TO 2);" "
;A$(3 TO 4);" ";A$(5 TO 6);
1130 LET DI=3
1135 GOTO 1180
1140 IF LEN A$<8 THEN GOTO 1025
1145 POKE NI,(16*CODE A$+CODE A$
(2)-476)
1150 POKE NI+1,(16*CODE A$(3)+CO
DE A$(4)-476)
1155 POKE NI+2,(16*CODE A$(5)+CO
DE A$(6)-476)
1160 POKE NI+3,(16*CODE A$(7)+CO
DE A$(8)-476)
1165 PRINT NI;" ";A$(1 TO 2);" "
;A$(3 TO 4);" ";A$(5 TO 6);" ";A
$(7 TO 8);
1170 LET DI=4
1180 PRINT "  C""EST OK ? "
1185 INPUT A$
1190 IF A$<>"" THEN GOTO 1230
1195 LET AI=NI
1200 LET NI=AI+DI
1210 GOTO 1020
1230 PRINT AT (23-PEEK 16442),0;
"ALORS ON RECOMMENCE"
1235 PRINT AT (22-PEEK 16442),31
;""
1240 GOTO 1025
1250 STOP
2000 INPUT A
2020 CLS
2025 FOR I=0 TO 21
2030 LET V=PEEK A
2035 PRINT A;TAB 10;V;TAB 16;CHR
$ V
2040 LET A=A+1
2045 NEXT I
2050 PAUSE 40000
2055 SCROLL
2060 LET V=PEEK A
2065 PRINT A;TAB 10;V;TAB 16;CHR
$ V
2070 LET A=A+1
2075 GOTO 2050
```

Le programme C.M.

CODES DE 16514 A 16635

16514	7	80	0	0	223
16519	0	237	60	0	0
16524	29	30	31	32	4
16529	4	0	36	37	0
16534	29	30	175	190	200
16539	120	129	71	62	155
16544	190	32	12	54	0
16549	229	204	75	65	0
16554	0	0	0	225	201
16559	62	1	185	192	62
16564	3	190	192	58	37
16569	64	254	247	204	209
16574	64	254	239	204	231
16579	64	201	128	128	126
16584	58	138	64	50	132
16589	64	201	0	0	56
16594	132	64	61	50	132
16599	64	203	127	32	5
16604	175	50	138	64	201
16609	62	255	50	138	64
16614	201	58	132	64	60
16619	50	132	64	254	0
16624	32	4	50	138	64
16629	201	62	1	50	138
16634	64	201			

CODES DE 16642 A 16741

16642	42	130	64	237	91
16647	132	64	25	203	123
16652	40	1	37	237	91
16657	134	64	25	203	123
16662	40	1	37	201	126
16667	128	56	134	64	237
16672	66	50	134	64	201
16677	128	128	56	136	64
16682	237	68	50	136	64
16687	58	132	64	237	66
16692	50	132	64	201	128
16697	128	205	39	65	205
16702	28	65	201	128	128
16707	43	124	181	32	251
16712	201	128	126	42	145
16717	64	35	34	145	64
16722	42	12	64	35	35
16727	35	35	126	254	165
16732	32	5	54	156	43
16737	24	246	60	119	201

CODES DE 16801 A 16895

16801	42	12	64	1	196
16806	2	9	34	136	64
16811	54	3	35	54	3
16816	35	54	3	0	0
16821	0	58	144	64	254
16826	1	56	6	61	50
16831	144	64	24	5	237
16836	75	145	64	201	42
16841	136	64	1	32	0
16846	175	237	66	54	52
16851	34	130	64	33	232
16856	3	205	67	55	42
16861	37	64	62	223	188
16866	32	7	62	127	189
16871	32	2	207	34	42
16876	130	64	1	1	0
16881	237	91	134	64	25
16886	203	123	40	1	37
16891	205	152	64	24	6

CODES DE 16902 A 17003

16902	237	91	138	64	175
16907	187	40	25	25	203
16912	123	40	1	37	14
16917	2	205	152	64	42
16922	130	64	25	203	123
16927	40	1	37	14	4
16932	205	152	64	120	254
16937	0	40	38	254	1
16942	204	28	65	254	2
16947	204	58	65	254	3
16952	204	28	65	254	4
16957	204	39	65	254	5
16962	204	56	65	254	6
16967	204	39	65	254	7
16972	204	58	65	195	220
16977	65	205	2	65	126
16982	254	0	196	200	64
16987	196	2	65	235	42
16992	130	64	54	0	235
16997	54	52	34	130	64
17002	24	6			

CODES DE 17010 A 17145

17010	237	91	12	64	175
17015	42	130	64	237	82
17020	17	214	2	237	82
17025	203	124	32	30	42
17030	130	64	54	0	33
17035	1	160	205	67	65
17040	42	136	64	62	2
17045	165	61	50	132	64
17050	50	138	64	205	186
17055	64	195	182	65	6
17060	2	205	173	66	16
17065	251	195	214	65	33
17070	232	3	205	67	65
17075	42	136	64	237	91
17080	12	64	175	0	237
17085	82	17	181	2	237
17090	82	62	2	189	48
17095	24	58	37	64	254
17100	247	32	17	42	136
17105	64	35	35	54	0
17110	43	43	43	54	3
17115	34	136	64	24	25
17120	125	254	29	48	20
17125	58	37	64	254	239
17130	32	13	42	136	64
17135	54	0	35	34	136
17140	64	35	35	54	3
17145	201				

```
>DISASSEMBLE £16536

4098 AF        XOR A
4099 BE        CP (HL)
409A C8        RET Z
409B 78        LD A,B
409C 81        ADD A,C
409D 47        LD B,A
409E 3E9B      LD A,9B
40A0 BE        CP (HL)
40A1 200C      JR NZ,40AF
40A3 3600      LD (HL),00
40A5 E5        PUSH HL
40A6 CC4B41    CALL Z,414B
40A9 00        NOP
40AA 00        NOP
>DISASSEMBLE £16555

40AB 00        NOP
40AC 00        NOP
40AD E1        POP HL
40AE C9        RET
40AF 3E01      LD A,01
40B1 B9        CP C
40B2 C0        RET NZ
40B3 3E03      LD A,03
40B5 BE        CP (HL)
40B6 C0        RET NZ
40B7 3A2540    LD A,(4025)
40BA FEF7      CP F7
40BC CCD140    CALL Z,40D1
40BF FEEF      CP EF
>DISASSEMBLE £16577

40C1 CCE740    CALL Z,40E7
40C4 C9        RET
40C5 80        ADD A,B
40C6 80        ADD A,B
40C7 80        ADD A,B
40C8 3A8A40    LD A,(408A)
40CB 328440    LD (4084),A
40CE C9        RET
40CF 00        NOP
40D0 00        NOP
40D1 3A8440    LD A,(4084)
40D4 3D        DEC A
40D5 328440    LD (4084),A
40D8 CB7F      BIT 7,A
>DISASSEMBLE £16602

40DA 2005      JR NZ,40E1
40DC AF        XOR A
40DD 328A40    LD (408A),A
40E0 C9        RET
40E1 3EFF      LD A,FF
40E3 328A40    LD (408A),A
40E6 C9        RET
40E7 3A8440    LD A,(4084)
40EA 3C        INC A
40EB 328440    LD (4084),A
40EE FE00      CP 00
40F0 2004      JR NZ,40F6
40F2 328A40    LD (408A),A
40F5 C9        RET
>DISASSEMBLE £16615

40E7 3A8440    LD A,(4084)
40EA 3C        INC A
40EB 328440    LD (4084),A
40EE FE00      CP 00
40F0 2004      JR NZ,40F6
40F2 328A40    LD (408A),A
40F5 C9        RET
40F6 3E01      LD A,01
40F8 328A40    LD (408A),A
40FB C9        RET
```

```
>DISASSEMBLE £16642

4102 2A8240    LD HL,(4082)
4105 ED5B8440  LD DE,(4084)
4109 19        ADD HL,DE
410A CB7B      BIT 7,E
410C 2801      JR Z,410F
410E 25        DEC H
410F ED5B8640  LD DE,(4086)
4113 19        ADD HL,DE
4114 CB7B      BIT 7,E
4116 2801      JR Z,4119
4118 25        DEC H
4119 C9        RET
>DISASSEMBLE £16668

411C 3A8640    LD A,(4086)
411F ED44      NEG
4121 328640    LD (4086),A
4124 C9        RET
4125 80        ADD A,B
4126 80        ADD A,B
4127 3A8A40    LD A,(408A)
412A ED44      NEG
412C 328A40    LD (408A),A
412F 3A8440    LD A,(4084)
4132 ED44      NEG
4134 328440    LD (4084),A
4137 C9        RET
4138 80        ADD A,B
>DISASSEMBLE £16698

413A CD2741    CALL 4127
413D CD1C41    CALL 411C
4140 C9        RET
4141 80        ADD A,B
4142 80        ADD A,B
4143 2B        DEC HL
4144 7C        LD A,H
4145 B5        OR L
4146 20FB      JR NZ,4143
4148 C9        RET
4149 80        ADD A,B
414A 80        ADD A,B
414B 2A9140    LD HL,(4091)
414E 23        INC HL
>DISASSEMBLE £16719

414F 229140    LD (4091),HL
4152 2A0C40    LD HL,(400C)
4155 23        INC HL
4156 23        INC HL
4157 23        INC HL
4158 23        INC HL
4159 7E        LD A,(HL)
415A FEA5      CP A5
415C 2005      JR NZ,4163
415E 369C      LD (HL),9C
4160 2B        DEC HL
4161 18F6      JR 4159
4163 3C        INC A
4164 77        LD (HL),A
>DISASSEMBLE £16741

4165 C9        RET
4166 00        NOP
4167 1D        DEC E
4168 1E1F      LD E,1F
416A 2021      JR NZ,418D
416C 222324    LD (2423),HL
416F 25        DEC H
4170 00        NOP
4171 1D        DEC E
4172 1E1F      LD E,1F
```

```
>DISASSEMBLE £16801

41A1 2A0C40    LD HL,(400C)
41A4 01C402    LD BC,02C4
41A7 09        ADD HL,BC
41A8 228840    LD (4088),HL
41AB 3603      LD (HL),03
41AD 23        INC HL
41AE 3603      LD (HL),03
41B0 23        INC HL
41B1 3603      LD (HL),03
41B3 00        NOP
41B4 00        NOP
41B5 00        NOP
41B6 3A9040    LD A,(4090)
41B9 FE01      CP 01
>DISASSEMBLE £16827

41BB 3806      JR C,41C3
41BD 3D        DEC A
41BE 329040    LD (4090),A
41C1 1805      JR 41C8
41C3 ED4B9140  LD BC,(4091)
41C7 C9        RET
41C8 2A8840    LD HL,(4088)
41CB 012000    LD BC,0020
41CE AF        XOR A
41CF ED42      SBC HL,BC
41D1 3634      LD (HL),34
41D3 228240    LD (4082),HL
41D6 21E803    LD HL,03E8
41D9 CD4341    CALL 4143
>DISASSEMBLE £16860

41DC 2A2540    LD HL,(4025)
41DF 3EDF      LD A,DF
41E1 BC        CP H
41E2 2007      JR NZ,41EB
41E4 3E7F      LD A,7F
41E6 BD        CP L
41E7 2002      JR NZ,41EB
41E9 CF        RST 08
41EA 222A82    LD (822A),HL
41ED 40        LD B,B
41EE 010100    LD BC,0001
41F1 ED5B8640  LD DE,(4086)
41F5 19        ADD HL,DE
41F6 CB7B      BIT 7,E
>DISASSEMBLE £16888

41F8 2801      JR Z,41FB
41FA 25        DEC H
41FB CD9840    CALL 4098
41FE 1806      JR 4206
```

```
>DISASSEMBLE £16902

4206 ED5B8A40 LD DE,(408A)
420A AF       XOR A
420B BB       CP E
>DISASSEMBLE £16908

420C 2819     JR Z,4227
420E 19       ADD HL,DE
420F CB7B     BIT 7,E
4211 2801     JR Z,4214
4213 25       DEC H
4214 0E02     LD C,02
4216 CD9840   CALL 4098
4219 2A8240   LD HL,(4082)
421C 19       ADD HL,DE
421D CB7B     BIT 7,E
421F 2801     JR Z,4222
4221 25       DEC H
4222 0E04     LD C,04
4224 CD9840   CALL 4098
>DISASSEMBLE £16935

4227 78       LD A,B
4228 FE00     CP 00
422A 2826     JR Z,4252
422C FE01     CP 01
422E CC1C41   CALL Z,411C
4231 FE02     CP 02
4233 CC3A41   CALL Z,413A
4236 FE03     CP 03
4238 CC1C41   CALL Z,411C
423B FE04     CP 04
423D CC2741   CALL Z,4127
4240 FE05     CP 05
4242 CC3A41   CALL Z,413A
4245 FE06     CP 06
>DISASSEMBLE £16967

4247 CC2741   CALL Z,4127
424A FE07     CP 07
424C CC3A41   CALL Z,413A
424F C3DC41   JP 41DC
4252 CD0241   CALL 4102
4255 7E       LD A,(HL)
4256 FE00     CP 00
4258 C4C840   CALL NZ,40C8
425B C40241   CALL NZ,4102
425E EB       EX DE,HL
425F 2A8240   LD HL,(4082)
4262 3600     LD (HL),00
4264 EB       EX DE,HL
4265 3634     LD (HL),34
>DISASSEMBLE £16999

4267 228240   LD (4082),HL
426A 1806     JR 4272

>DISASSEMBLE £17010

4272 ED5B0C40 LD DE,(400C)
4276 AF       XOR A
4277 2A8240   LD HL,(4082)
427A ED52     SBC HL,DE
427C 11D602   LD DE,02D6
427F ED52     SBC HL,DE
4281 CB7C     BIT 7,H
4283 201E     JR NZ,42A3
4285 2A8240   LD HL,(4082)
4288 3600     LD (HL),00
428A 2101A0   LD HL,A001
428D CD4341   CALL 4143
4290 2A8840   LD HL,(4088)
4293 3E02     LD A,02
>DISASSEMBLE £17045

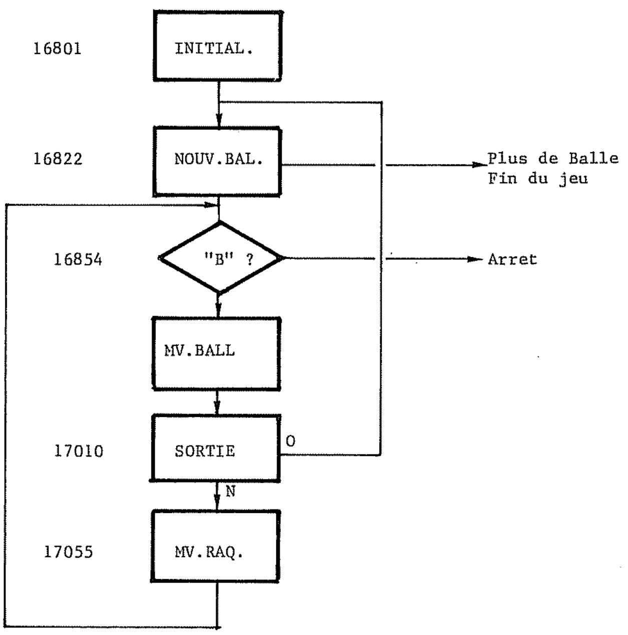
4295 A5       AND L
4296 3D       DEC A
4297 328440   LD (4084),A
429A 328A40   LD (408A),A
429D CD8A40   CALL 408A
42A0 C3B641   JP 41B6
42A3 0602     LD B,02
42A5 CDAD42   CALL 42AD
42A8 10FB     DJNZ 42A5
42AA C3D641   JP 41D6
42AD 21E803   LD HL,03E8
42B0 CD4341   CALL 4143
42B3 2A8840   LD HL,(4088)
42B6 ED5B0C40 LD DE,(400C)
>DISASSEMBLE £17082

42BA AF       XOR A
42BB 00       NOP
42BC ED52     SBC HL,DE
42BE 11B502   LD DE,02B5
42C1 ED52     SBC HL,DE
42C3 3E02     LD A,02
42C5 BD       CP L
42C6 3018     JR NC,42E0
42C8 3A2540   LD A,(4025)
42CB FEF7     CP F7
42CD 2011     JR NZ,42E0
42CF 2A8840   LD HL,(4088)
42D2 23       INC HL
42D3 23       INC HL
>DISASSEMBLE £17108

42D4 3600     LD (HL),00
42D6 2B       DEC HL
42D7 2B       DEC HL
42D8 2B       DEC HL
42D9 3603     LD (HL),03
42DB 228840   LD (4088),HL
42DE 1819     JR 42F9
42E0 7D       LD A,L
42E1 FE10     CP 10
42E3 3014     JR NC,42F9
42E5 3A2540   LD A,(4025)
42E8 FEEF     CP EF
42EA 200D     JR NZ,42F9
42EC 2A8840   LD HL,(4088)
>DISASSEMBLE £17135

42EF 3600     LD (HL),00
42F1 23       INC HL
42F2 228840   LD (4088),HL
42F5 23       INC HL
42F6 23       INC HL
42F7 3603     LD (HL),03
42F9 C9       RET
```

CASSE-BRIQUES ORGANIGRAMME



. INIT : Initialise et dessine la raquette.
. NOUV. BAL : Vérifie s'il reste des balles. Si non, fin du jeu. Si oui, dessine la nouvelle balle sur la raquette.
. "B" : Interrompt le programme machine par appui sur touche B.
. MV. BAL : Sonde la position future de la balle (présence de brique).
Efface les briques touchées (routine TEST).
Tient à jour le score (routine COMPT).
Assure les rebonds (routines $\overline{DL}$, $\overline{DC}$, $\overline{DLC}$, EFFET).
Assure le déplacement de la balle (routines NV.ADR. NORM).
. SORTIE : Vérifie si la balle est sortie du terrain.
. MV.RAQ : Déplace la raquette si touche 1 ou 0 pressée (routine D.RAQ).

LES GRANDES LIGNES DU JEU

Les limites du terrain et les briques sont affichées à partir du programme BASIC.
La raquette et la balle sont affichées par le programme C.M., leurs positions étant connues par leurs adresses (dans le fichier d'affichage), elles-mêmes sauvegardées dans les octets :

Balle : 16514/15
Raquette (élément gauche) : 16520/21

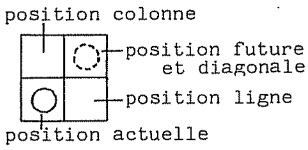
Le déplacement de la balle, pour chaque cycle, est défini par deux valeurs :

DC, vitesse horizontale : nombre de colonnes parcourues,
DL, vitesse verticale : nombre de lignes parcourues x 33.

Si on appelle A l'adresse actuelle de la balle, l'adresse future sera A'= A + DC + DL.

Avant chaque déplacement de la balle, on regarde quels obstacles sont touchés dans le sens de son mouvement.

Exemple : DC = +1
DL = -1



Si les obstacles touchés forment une ligne horizontale, le rebond se fera en chargeant le signe de DL.
Si les obstacles touchés forment une ligne verticale, le rebond se fera en chargeant le signe de DC.
Si les obstacles forment un coin, rentrant ou saillant, la balle reviendra sur ses pas.

Les briques touchées sont effacées et le compteur de score incrémenté, mais tout caractère autre qu'un espace sera considéré comme un obstacle, donc produira un rebond. Si de plus le caractère "touché" correspond à la raquette, la vitesse horizontale est incrémentée ou décrémentée de 1 colonne selon le sens de déplacement de celle-ci.

Si l'adresse de la balle correspond à une ligne située sous la raquette, elle est considérée comme sortie, s'efface et une nouvelle balle est produite. Après la 5ème balle, le jeu est terminé et rend le contrôle au BASIC.

La raquette se déplace si les touches "1" ou "0" sont reconnues. Afin de laisser quelques chances au joueur, elle se déplace 2 fois plus vite que la balle.

Adresse	C.M.	Mnémonique	
		TABLE DES REM	
1 REM	122 car.	de 16514 à 16635	
2 REM	153 car.	de 16642 à 16794	
3 REM	95 car.	de 16801 à 16895	
4 REM	102 car.	de 16902 à 17003	
5 REM	136 car.	de 17010 à 17145	
		VARIABLES ET PARAMETRES	
16514/15		Adresse de la balle	
16516/17		DC, vitesse horizontale (-2,-1,0,+1 ou +2)	
16518/19		DL, vitesse verticale x 33 (±33)	
16520/21		Adresse raquette	
16522/23		DCN (= SGN DC)	
16524/27		libre	
16528		Balles restant à jouer	
16529/30		Score du jeu en cours	
16531/35		libre	
16536		Début 1ère routine	

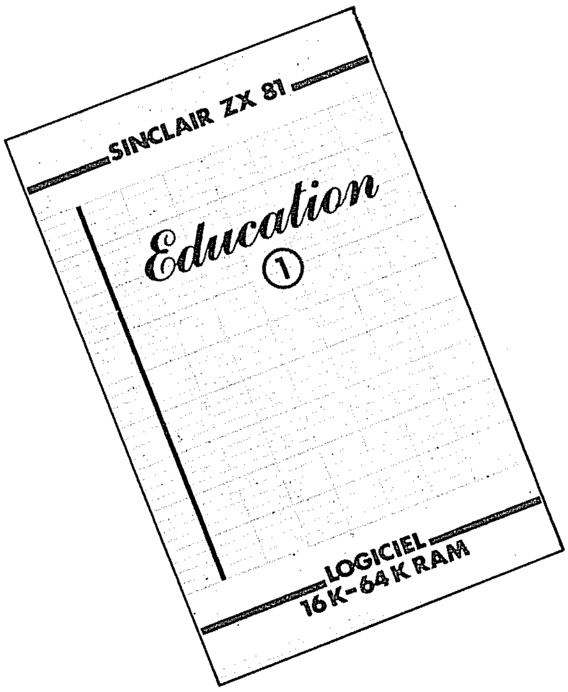
DEMONTAGE DU PROGRAMME

- INIT, 16801/19 : La raquette est située au début au milieu de la ligne 21. L'adresse de son élément gauche est sauvegardée, puis on affiche 3 caractères graphiques ▀ (code 03).
- NOUV.BAL
  . 16822 à 16839 : vérifie s'il reste au moins une balle en réserve (octet 16528). Si oui, décrémente le contenu de la réserve et passe à la suite. Si non, rend le contrôle au BASIC.
  . 16840 à 16853 : calcule l'adresse de la balle pour la situer sur la raquette, et l'affiche, et sauvegarde son adresse.
- MV.BAL
  . 16854/59 : règle la rapidité du jeu par la routine DELAÏ, qui crée une pause dont la durée dépend de HL.
  . 16860/74 : permet l'interruption du programme si la touche "B" est reconnue.
  . 16875/93 : après avoir initialisé B à 0 et C à 1, calcule l'adresse de la position colonne en ajoutant DL à l'adresse de la balle, et appelle la routine TEST pour la position colonne.
  . 16894/95 : passage au REM suivant.
  . 16902/909 : évite de tester les positions ligne et diagonale pour un déplacement vertical (DC donc le registre E est alors à 0).
- D.RAQ :
  . 17069/74 : petite pause avant chaque déplacement de la raquette, pour éviter un trop grand effet de saccade.
  . 17075/90 : à partir de l'adresse, recalcule la position de l'élément gauche de la raquette sur sa ligne.
  . 17091/94 : limite le déplacement à gauche.
  . 17096/17119 : déplace la raquette d'une colonne à gauche si la touche "1" est reconnue.
  . 17120/145 : idem p. déplacement à droite (butée et touche "0").
- SUITE
  . 16935/77 : modifie la vitesse de la balle selon la valeur du registre B, qui dépend de la configuration des obstacles rencontrés :
  B = 0 : pas d'obstacles
  B ≠ 0 (1 à 7) : changement du signe de DL, DC ou des deux par appel aux routines $\overline{DL}$, $\overline{DC}$ ou $\overline{DLC}$.
- DEP.BAL.
  . 16978/89 : calcule la nouvelle adresse de la balle compte tenu des composantes de sa vitesse après rebond. Pour éviter des effacements intempestifs d'obstacles (pour des vitesses DC > 1), la vitesse DC est normée si nécessaire.
  . 16990/17001 : déplace effectivement la balle
  . 17002/3 : passe au REM suivant.
- SORTIE
  . 17010/28 : vérifie si l'adresse de la balle est ou non au-dessous de la ligne 21. Si non passe à MV.RAQ.
  Si oui, la la balle est sortie. Dans ce cas :
  . 17029/65 : efface la balle, et après une attente redéfinit la nouvelle vitesse de la balle avec un certain élément de hasard selon la position de la raquette, puis revient à NV.BAL. (nouvelle balle).
- MV.RAQ. appelle 2 fois D.RAQ. avant de sauter à MV. BAL. En pokant 1, 2 ou 4 à l'adresse 17060, la raquette se déplacera 1, 2 ou 4 fois plus vite que la balle
  . 16910/20 : calcule l'adresse de la position diagonale avant d'appeler TEST. C est chargé de la valeur 2.
  . 16921/34 : idem pour la position ligne, mais avec la valeur 4 pour C.
- Routine TEST
  . 16536/41
  a) revient au programme principal si le contenu de l'adresse testée est nul (pas d'obstacle).
  b) ajoute C à B dans le cas contraire. Ainsi après avoir testé les 3 positions, B contiendra une valeur de 0 à 7 qui permettra de définir le type de rebond à effectuer.
  . 16542/66 : dans ce cas, la balle touche un obstacle. Ce peut être une brique, le bord du terrain ou la raquette.
  S'il s'agit d'une brique, elle est effacée et le score est incrémenté par appel à la routine COMPT. Sinon, s'il s'agit à la fois d'une position colonne et de la raquette, on continue sur EFFET.
- EFFET, 16567/80 : reconnaît les touches 1 et 0 et fait appel si nécessaire à EFG ou EFD (effet à gauche ou à droite).
- EFG, 16593/16614 : décrémente la vitesse horizontale d'une unité : si la balle va vers la gauche, elle ira plus vite, si elle va à droite, elle ira moins vite.
- EFD, 16815/35 : idem pour l'effet à droite.

-NORM : ramène à ±1 la vitesse horizontale.

B. Lacoste

Trois utilitaires ZX81



ORDI-5 vous donne rapidement son avis sur trois cassettes utilitaires destinées aux programmeurs confirmés qui font honneur à ce qu'on appelle, dans les milieux technocratiques "le génie logiciel" français.

EDUCATION SINCLAIR 16K-64K DE RAM 100 F

Voici une cassette assez originale dans son objet et à laquelle nous ne connaissons pas d'équivalent sur le marché. Beaucoup d'enseignants ou de formateurs sont très intéressés par les possibilités pédagogiques des petits ordinateurs surtout lorsqu'ils ne sont pas très coûteux et peuvent dont être fournis en plusieurs exemplaires à une classe plus ou moins nombreuse. Malheureusement la construction d'un didacticiel, c'est-à-dire d'un programme orienté sur la pédagogie, implique que son auteur soit un programmeur confirmé afin de tirer le meilleur parti de l'organisation mémoire et trouver les procédures les plus souples ; enfin il est nécessaire que la programmation soit structurée pour éviter tout plantage de l'élève.

La cassette " éducation" de Sinclair vient à l'aide des formateurs qui souhaitent construire des didacticiels sans pour cela programmer superbement en BASIC. Cette cassette est en quelque sorte une aide à la construction de didacticiels. Elle demande à l'utilisateur, que la documentation appelle très justement l'"initiateur", quel est le type d'exercice qu'il veut programmer, comment vont se répartir les différents textes, les écrans, quel est le mode choisi (questions-réponses) et le logiciel se structure lui-même en fonction des options choisies.

Ce programme a une portée très générale et la possibilité de l'adapter au matériel choisi 16 ou 64 K lui donne encore un attrait supplémentaire. Il est en particulier, dans la version 64K, possible d'arriver à des compositions d'écran très élaborées. Le seul inconvénient est que la démarche, lourde et rigide, implique une grande attention de la part de l'initiateur et probablement un bon nombre de tâtonnements pour découvrir la meilleure formule, d'autant que la notice est assez dense et assez peu vivante (c'est évidemment inévitable lorsqu'il faut faire état d'un grand nombre de possibilités) : de toutes façons, à partir du moment où l'on se lance dans la construction de didacticiels, c'est que l'on a pris une option sur des applications sérieuses et que par conséquent on est prêt à travailler. Il ne faut pas se laisser rebuter par l'aspect un peu ingrat de cette cassette au premier abord : on peut la considérer comme un utilitaire de la meilleure catégorie. Nous aimerions assez entendre parler d'expériences à l'aide de ce programme.

ZX TRI SINCLAIR ZX81 16K-64K DE RAM 100 F

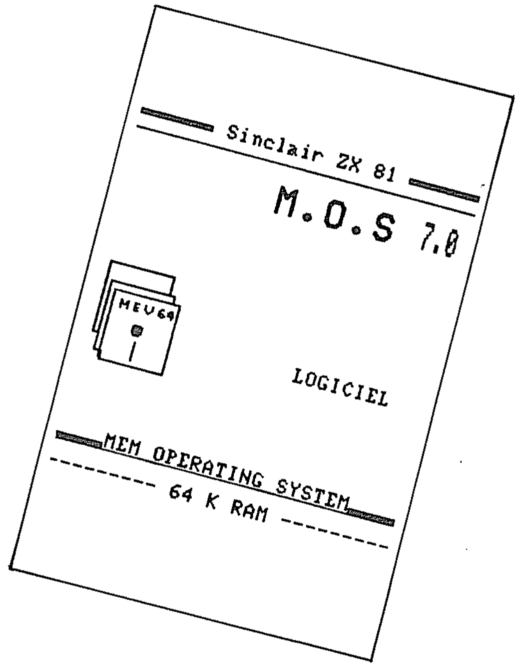
Nous n'avons pas peur du pléonasme, le ZX TRI est l'un des utilitaires les plus utiles que nous ayons rencontrés pour le ZX81, du moins en ce qui concerne la construction de programmes de fichier. Entendons-nous bien : le ZX TRI n'est pas un programme de fichier au sens où on l'entend habituellement. Les programmes de fichiers ordinaires présentent un certain nombre de rigidités qui limitent considérablement leur usage : qu'ils soient spécialisés ou que, d'une façon plus sophistiquée, ils proposent à l'utilisateur de définir lui-même ses propres champs, ces programmes sont intangibles : surtout écrits en langages machine, ils reposent sur une organisation stricte des différentes routines (insertions, recherche, correction) au coeur desquelles l'utilisateur ne peut pas intervenir sans tout bouleverser.



Avec le ZX TRI on a affaire à une toute autre philosophie : on attend de l'utilisateur qu'il se construise lui-même son programme de fichier, mais s'ajoutent, à la batterie des instructions BASIC habituelles, 5 mots nouveaux pratiquement utilisables comme les mots du dictionnaire de base, chacun de ces mots correspondant à une fonction extrêmement utile et puissante d'insertion, de recherche, de suppression d'un élément de tri. La simple utilisation de ces mots : INS, DEL, REC, avec des paramètres correctement choisis appelle des routines machine qui se chargent des additions, soustractions, et décalages. Ces différents sous-programmes seraient longs à écrire et surtout à exécuter en BASIC, même avec intervention sur les variables-système : l'incrémentation d'un tableau de N éléments déclarés à N + 1 éléments, en cas d'insertion par exemple, n'est pas à la portée du premier programmeur venu et nécessite en tous cas une parfaite connaissance de la disposition de la mémoire. Grâce à ces instructions, simplement introduites dans une REM et précédés d'un RAND USR 16516 il est possible de réaliser des programmes de fichier étonnamment courts et rapides et comportant l'essentiel des procédures nécessaires dans ce type d'application. Les seules difficultés, mineures, que nous ayons rencontrées concernent l'édition lorsque l'on choisit la version 64K. En définitive une belle réalisation très souple qui laisse entièrement l'initiative au programmeur.

MOS 7.0 UN OPERATING SYSTEME POUR ZX81 - MVI 100 F

L'opérating système d'un ordinateur est sommairement défini comme le programme en langage machine qui permet de gérer les fichiers sur disquettes. Comment le ZX81, qui ne peut être muni de disquettes (pour l'instant !) peut-il supporter un opérating système ? Il s'agit en fait plutôt d'un simulateur d'opérating système sur mémoire vive : en travaillant sous ce logiciel, l'utilisateur a l'impression de travailler comme avec une disquette ; il peut ainsi écrire différents programmes même rédigés aveç des numérés de lignes identiques et les envoyer dans la mémoire sans se préoccuper de l'endroit où ils seront stockés, le logiciel le fera pour lui. L'ensemble des commandes que l'on trouve pour la gestion de disquettes et ainsi présent : FORMAT, pour préparer la mémoire à recevoir des programmes calibrés, CAT pour lire instantanément l'ensemble des fichiers de la mémoire, etc. Cet o.s. permet de faire des chaînages entre différents programmes, de les renumérotés, de les détruire sélectivement, etc. C'est une grande statisfaction de pouvoir travailler sous une dizaine de programmes différents, et même plus, sans avoir à passer et repasser par les opérations fastidieuses de sauvegarde et chargement cassettes.



Les va-et-vient se font instantanément. La programmation est confortable et on se sent vraiment travailler "sous" O.S.

Ce logiciel n'occupe que 2K de mémoire, ce qui permet de la placer en différents endroits et donc de le rendre compatible avec d'autres softs performants (FLM ou ZX TRI). Nous ne voyons pas d'autre alternative pour ceux qui souhaitent travailler en ayant une quinzaine de programmes à leur disposition immédiate. Cette belle réalisation souffre de petites limitations que nous nous n'avons pas sû contourner facilement : si l'on veut chaîner des programmes il faut les avoir tous directement transcrit au clavier ; l'O.S. ne gère que la zone programme les variables ne peuvent passer de programme à programme qu'à l'aide des artifices habituels.

B. T.

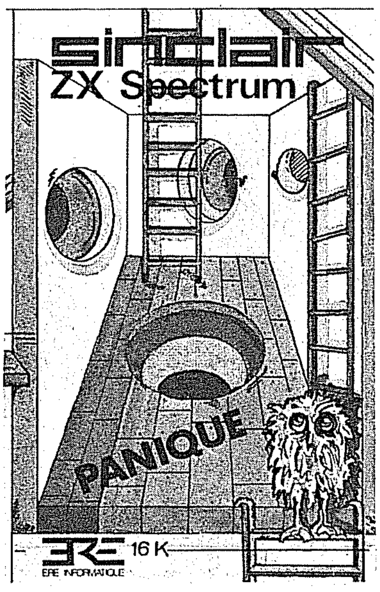
5 Cinq cassettes spectrum

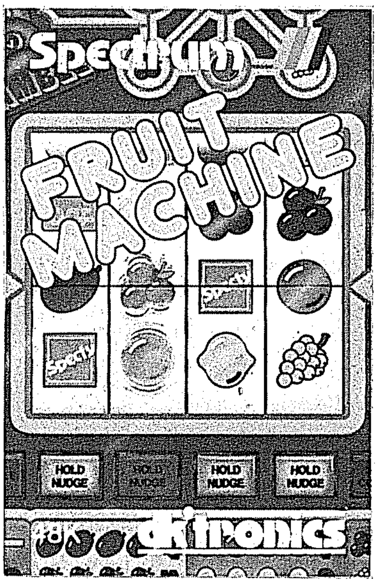
PANIQUE (ERE INFORMATIQUE 1983) 16 K OU 48 K - 75 F

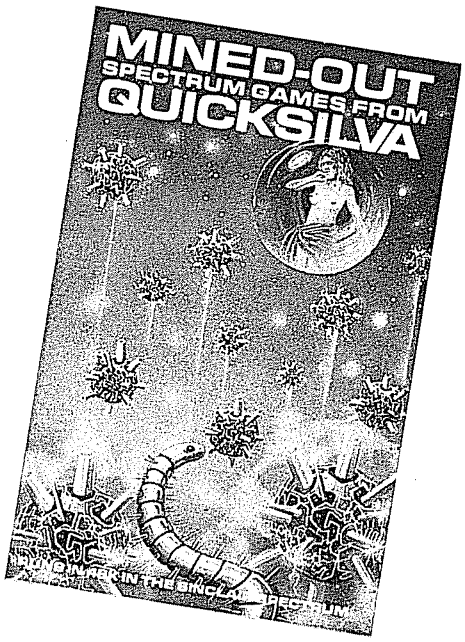
Ere Informatique a réussi là un joli jeu très coloré et au graphisme parfaitement clair. Les auteurs annoncent qu'il s'agit d'un jeu d'arcade et effectivement il n'est pas sans rappeler le jeu bien connu dans les cafés où les gorilles précipitent des tonneaux sur des bonhommes qui disposent des échelles, etc...
Le propos est plus familier puisque tout se passe de la cave au grenier dans une maison et qu'il s'agit simplement d'éviter qu'un animal qui ressemble à une grosse chouette ne mange une provision de choux. Le nombre de décisions à prendre simultanément est important et, spécialement pour ceux qui n'ont pas de manette, il est difficile de gérer les touches curseur simultanément pour lever une échelle, percer un trou ou aller et venir. Jeu difficile mais franchement amusant, cependant l'absence de variété se fait sentir à la longue. Rapidité moyenne.

MINED OUT - (QUICKSILVA 83) - 16 OU 48 K - 80 F

Un jeu dont on se lasse fificilement bien que le principe en soit très simple : il faut progresser dans un champ de mines dont on ne connaît pas l'emplacement pour atteindre l'autre extrémité du champ. On peut être distrait de sa route par quelques demoiselles éplorées. Un indicateur de proximité précise le nombre de mines qui menacent directement et il faut alors revenir sur ses pas pour repartir dans une autre direction. Les niveaux se succèdent en difficulté. Une fois la passe terminée, le cheminement suivi se reconstitue, cette fois avec les mines en évidence. Il s'agit d'une belle réalisation en BASIC de rapidité moyenne mais d'un réel intérêt.







FUIT MACHINE (DK TRONICS1983) 16 OU 48 K - 75 F

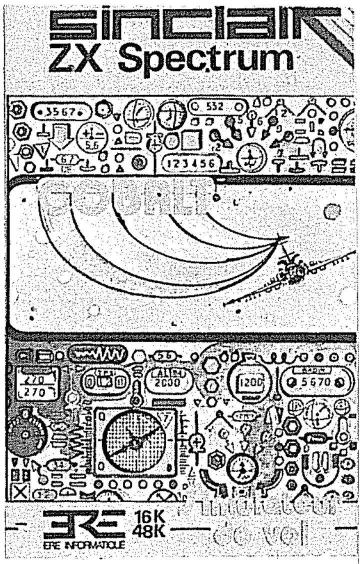
L'un des plus beaux jeux de chez DK'Tronics : un graphisme réellement époustouflant pour servir un jeu de Jack-Pot assez classique. Il faut noter la très grande rapidité dans le défilement des figures (oranges, citrons, etc) dont le réalisme en dit long sur les étonnantes possibilités du Spectrum dans le domaine graphique. Nous sommes restés très admiratifs devant la musique d'accompagnement spirituelle. C'est une véritable fête sur l'écran. Il faut s'accrocher pour savoir ce que l'on fait (montant des paris, etc.), et faire très attention à ne pas toucher "BREAK" car alors le jeu s'auto-efface. Un gros défaut toutefois, qu'il devrait être possible de corriger : les instructions du programme ne sont pas francisées, or elles sont assez complexes.

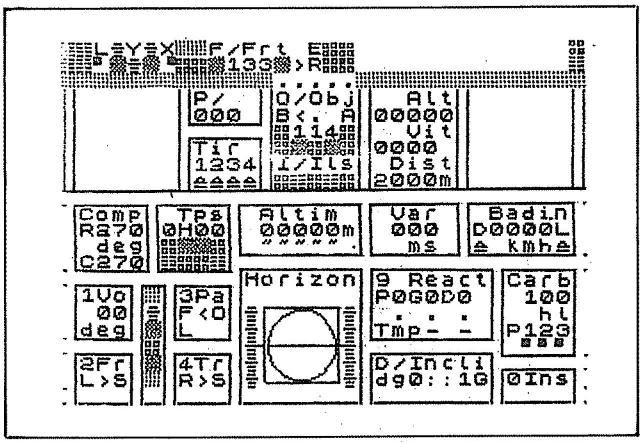
TERRORDACTYL - (MELBOURNE HOUSE) - 48 K - ENV. 100 F



C'est une variante intéressante des fameux "invaders" qui se démarque des autres jeux de ce genre par la qualité du graphisme ambiant, ce qui change des figures géométriques de ce jeu peu trop souvent vu. L'argument est le suivant : votre avion s'écrase dans une jungle inconnue et dans ce décor hostile les envahisseurs sont des espèces (!) de ptérodactyles préhistoriques très difficiles à écarter avec un simple fusil : l'un deux finit toujours par fondre sur vous d'un seul coup, et là, c'est assez effrayant. Décor très riche constitué d'une montagne, de rivières, etc... et naturellement de la carcasse de l'avion ; une certaine monotonie toutefois due au jeu lui-même.

INTERCEPTEUR COBALT (ERE INFORMATIQUE) 16 OU 48 K - 95F





Enfin un très beau jeu interactif complet entièrement en français : près de 45 commandes rendent ce simulateur de vol particulièrement performant et même un modèle du genre visiblement très documenté. Ce type de jeu requiert énormément de concentration et la faculté de pouvoir fixer très attentivement un tableau. Nous reproduisons ci-dessous un bel écran.

Ph. D.

l'ordinateur en classe
Le ZX fort en thème



L'ordinateur fort en thème ? Ou encore, quel est le génitif de Zédix ? Zedivis ? Zedinis ? Soyons sérieux, avec J.C. Hein qui nous propose d'avoir un pied dans le passé et un pied dans l'avenir avec ce beau programme de déclinaison latine.

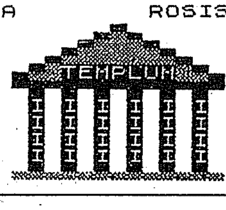
```
   1 REM     ******************
             *    J. C. HEIN  *
             *    (C) 1983    *
             ******************
   2 REM
   3 REM
   4 GOSUB 9000
   5 GOTO 12
   9 STOP
  10 CLS
  12 REM DECLINAISONS LATINES
  13 LET ZX=0
  14 LET PA=8
  15 CLS
  20 PRINT "       DECLINAISONS LA
TINES"
  30 PRINT
  40 PRINT
  50 PRINT "NOMINATIF DU MOT ?"
  60 INPUT N$
  70 IF LEN N$<3 AND N$<>"OS" TH
EN GOTO 60
  72 PRINT N$
  76 PRINT
  80 PRINT "GENITIF DU MOT ?"
  83 INPUT Q$
  84 IF LEN Q$<3 THEN GOTO 83
  88 PRINT Q$
  90 LET R$=Q$( TO (LEN Q$-2))
  95 LET G$=Q$((LEN Q$-1) TO )
 100 IF G$="AE" OR G$="IS" OR G$
="US" OR G$="EI" THEN GOTO 120
 110 LET R$=Q$( TO (LEN Q$-1))
 113 LET G$=Q$((LEN Q$) TO .)
 117 IF G$<>"I" THEN GOTO 10
 118 REM ***********************
 120 IF Q$="ITINERIS" THEN GOTO
554
 122 IF Q$="HOMINIS" THEN GOTO 5
50
 124 IF Q$="PATRIS" OR Q$="MATRI
S" OR Q$="FRATRIS" OR Q$="JUVENI
S" OR Q$="SENIS" OR Q$="CANIS" T
HEN GOTO 560
 128 IF G$="AE" THEN GOTO 180
 130 IF G$="I" THEN GOTO 227
 140 IF G$="IS" THEN GOTO 310
 150 IF G$="US" THEN GOTO 650
 160 IF G$="EI" THEN GOTO 720
 170 REM *********************
 180 REM ********PREMIERE*****
 200 LET A$="A       AE      A       A
E      AM      AS      AE      ARUM    AE
   IS      A       IS      "
 205 CLS
 210 PRINT TAB 5;"PREMIERE DECLI
NAISON"
 215 PRINT
 217 LET V$=N$
 218 GOTO 1000
 220 REM ***********************.
 225 REM *****DEUXIEME**********
 227 LET A$="US      I       E       I
       UM      OS      I       ORUM    O
   IS      O       IS      "
 228 IF N$((LEN N$-1) TO )="UM"
THEN GOTO 250
 230 IF N$((LEN N$) TO )="R" THE
N GOTO 275
 231 CLS
```



```
 232 PRINT TAB 5;"DEUXIEME DECLI
NAISON"
 234 GOTO 1000
 250 LET A$="UM     A      UM     A
     UM     A      I      ORUM   O
   IS     O      IS     "
 255 CLS
 260 PRINT " DEUXIEME DECLINAISO
N (NEUTRE)"
 265 PRINT
 270 GOTO 1000
 275 LET R$=Q$( TO (LEN Q$-1))
 277 LET ZX=1
 280 LET A$="           I           I
     UM     OS     I      ORUM   O
   IS     O      IS     "
 285 CLS
 290 PRINT TAB 5;"DEUXIEME DECLI
NAISON"
 295 PRINT
 300 GOTO 1000
 310 REM *********************
 320 REM ******TROISIEME*******
 330 REM PARISYLLABIQUE ?
 340 IF N$(1 TO (LEN R$))<>R$ TH
EN GOTO 344
 342 GOTO 350
 344 LET PA=0
 346 GOTO 380
 350 LET A$=N$((LEN R$+1) TO  )
 352 IF LEN A$=0 OR LEN A$>2 THE
N GOTO 356
 354 GOTO 360
 356 LET PA=0
 358 GOTO 380
 360 IF A$="E" OR A$="ES" OR A$=
"IS" THEN LET PA=1
 370 REM -----FAUX IMPARIS.-----
 380 IF LEN R$<2 THEN GOTO 384
 382 GOTO 390
 384 LET FP=0
 386 GOTO 420
 390 LET H$=R$(LEN R$-1)
 393 LET L$=R$(LEN R$)
 396 LET FP=0
 400 IF H$="A" OR H$="E" OR H$="
I" OR H$="O" OR H$="U" OR H$="Y"
 THEN GOTO 420
 410 IF L$<>"A" AND L$<>"E" AND
L$<>"I" AND L$<>"O" AND L$<>"U"
AND L$<>"Y" THEN LET FP=1
 415 REM GENRE?
 420 PRINT
 425 PRINT "GENRE DU MOT ?"
 427 PRINT
 428 PRINT "      -MASCULIN:1
                  -FEMININ:2
                  -NEUTRE:3
           "
 429 PRINT "ENTREZ LE NUMERO COR
RESPONDANT."
 430 INPUT Z$
 435 IF Z$<>"1" AND Z$<>"2" AND
Z$<>"3" THEN GOTO 430
 440 REM FAUX IMPARISYLLABIQUES
 470 IF Z$="3" AND (N$((LEN N$-1
) TO  )="AR" OR N$((LEN N$-1) TO
  )="AL") THEN LET PA=1
 480 REM DESINENCES
 490 IF Z$="3" AND PA=1 THEN GOT
O 495
 492 GOTO 510
 495 LET A$="E          IA     E      I
A    E      IA     IS     IUM    I
   IBUS   I      IBUS   "
 497 LET ZX=2
 500 CLS
 502 PRINT " 3 EME:PARISYLLABIQU
ES NEUTRES"
 504 PRINT
 506 GOTO 1000
 510 REM PARI+FAUX PARI
 520 IF FP=1 OR PA=1 THEN GOTO 5
25
 522 GOTO 540
 525 LET A$="IS     ES     IS     E
S    EM     ES     IS     IUM    I
   IBUS   E      IBUS   "
 527 LET ZX=1
 528 CLS
 532 PRINT
 535 PRINT "  3 EME:PARI. ET FAU
X IMPARI."
 537 GOTO 1000
 540 REM IMPARISSYLLAB. NEUTRE
 550 IF Z$="3" THEN GOTO 554
 552 GOTO 560
 554 LET A$="           A           A
            A      IS     UM     I
   IBUS   E      IBUS   "
 555 LET ZX=2
 556 CLS
 557 PRINT "3 EME:IMPARISYLLABIQ
UES NEUTRES"
 558 PRINT
 559 GOTO 1000
 560 LET A$="           ES          E
S    EM     ES     IS     UM     I
   IBUS   E      IBUS   "
 562 LET ZX=1
 565 CLS
 570 PRINT "     3 EME:IMPARISYLL
ABIQUES"
 575 PRINT
 580 GOTO 1000
 650 REM *********************
 660 REM *******QUATRIEME*******
 670 LET A$="US     US     US     U
S    UM     US     US     UUM    UI
   IBUS   U      IBUS   "
 675 IF N$(LEN N$)="U" THEN GOTO
 700
 680 CLS
 685 PRINT TAB 5;"QUATRIEME DECL
INAISON"
 690 PRINT
 695 GOTO 1000
 700 LET A$="U      UA     U      U
A    U      UA     US     UUM    UI
   IBUS   U      IBUS   "
 705 CLS
 710 PRINT " QUATRIEME DECLINAIS
ON (NEUTRE)"
 713 PRINT
 717 GOTO 1000
 720 REM *********************
 725 REM *****CINQUIEME******
 730 LET A$="ES     ES     ES     E
S    EM     ES     EI     ERUM   EI
   EBUS   E      EBUS   "
 740 CLS
 750 PRINT TAB 5;"CINQUIEME DECL
INAISON"
 760 PRINT
 997 REM
 998 REM
 999 REM *********************
1000 PRINT
1010 PRINT "CAS:   SINGULIER:   PL
URIEL:"
1020 PRINT """"""""   """"""""""""
""""""""""    """""""""""""""""
1030 LET X=1
1035 LET Y=1
1040 LET W$="NOM.VOC.ACC.GEN.DAT
.ABL."
1050 FOR D=6 TO 16 STEP 2
1060 PRINT AT D,0;W$(X TO X+3)
1070 LET X=X+4
1075 PRINT AT D,6;R$+A$(Y TO Y+5
)
1080 LET Y=Y+6
1085 IF ZX=1 OR ZX=2 THEN PRINT
AT 6,6;N$;"    ";AT 8,6;N$;"    "
1087 IF ZX=2 THEN PRINT AT 10,6;
N$;"    "
1090 PRINT AT D,18;R$+A$(Y TO Y+
5)
1100 LET Y=Y+6
1110 NEXT D
1113 LET ZX=0
1115 REM *********************
1116 REM *********************
1120 PRINT AT 20,0;"VOULEZ-VOUS
RECOMMENCER ? (O/N)"
1130 INPUT S$
1140 IF S$<>"O" AND S$<>"N" THEN
 GOTO 1130
1150 IF S$="N" THEN GOTO 9
1160 GOTO 10
9000 PRINT AT 5,5;"[graphic characters]"
9010 PRINT TAB 5;"[graphic characters]"
9020 PRINT TAB 5;"[graphic characters]"
9030 PRINT TAB 5;"[graphic characters]"
9040 PRINT TAB 5;"[graphic characters]"
9050 PRINT TAB 5;"[graphic characters]"
9070 PRINT AT 12,16;"[graphic characters]"
9075 PRINT AT 13,14;"[graphic characters]"
9080 PRINT AT 14,12;"[graphic characters]"
9085 PRINT AT 15,10;"[graphic characters] TEMPLUM [graphic characters]"
9100 PRINT AT 16,10;"[graphic characters]"
9110 PRINT AT 17,11;"[graphic characters]"
9112 PRINT AT 18,11;"[graphic characters]"
9114 PRINT AT 19,11;"[graphic characters]"
9116 PRINT AT 20,11;"[graphic characters]"
9118 PRINT AT 21,10;"[graphic characters]"
9200 FOR N=0 TO 60
9210 NEXT N
9220 RETURN
```

PREMIERE DECLINAISON

CAS:	SINGULIER:	PLURIEL:
NOM.	ROSA	ROSAE
VOC.	ROSA	ROSAE
ACC.	ROSAM	ROSAS
GEN.	ROSAE	ROSARUM
DAT.	ROSAE	ROSIS
ABL.	ROSA	ROSIS



J.C. Hein

Programmes de jeu

BLINDES (ZX81 16K)

Nous allons sacrifier dans ce numéro à la vogue des "wargames". Dans cette bataille de chars, qui se déroule très lentement, il faut anticiper les mouvements de l'adversaire et donner des instructions chiffrées à ses propres unités : attention, un "wargame" n'est pas un jeu interactif, on est plutôt comme dans un état-major à voir évoluer des maquettes sur une grande carte ; ce jeu implique une certaine capacité de concentration et d'abstraction.

```
   1 RAND
   5 CLEAR
   7 CLS
  10 PRINT AT 0,11;"▓       ▓"
  20 PRINT AT 1,11;"▓▓▓▓▓▓▓▓▓"
  30 PRINT AT 2,11;"▓KOBRA▓▓"
  35 PRINT AT 3,11;"▓▓▒▒▒▓▓▓"
  40 PRINT AT 4,11;"▓▓▓▓▓▓▓▓"
  45 PRINT AT 5,11;"▓< H >▓"
  50 PRINT AT 6,11;"▓▓▓H▓▓▓"
  55 PRINT AT 7,11;"▓▓▓H▓▓▓"
  60 PRINT AT 8,11;" ▓▓H▓▓"
  65 PRINT AT 9,11;"  ▓H▓"
  70 PRINT AT 10,11;"   ▓"
  75 PRINT AT 11,11;"   ▓"
 100 FOR Z=1 TO 29
 110 PRINT AT 13,8;"C O M B A T III"
 120 PRINT AT 13,8;"C O M B A T
III"
 125 PRINT AT 16,Z;"  ,"
 130 PRINT AT 17,Z;" ▓▓ "
 135 PRINT AT 20,5;"COPYRIGHT CH
OUPS 1983"
 140 NEXT Z
 142 FOR R=12 TO 21 STEP .25
 143 PRINT AT R,14;"▓"
 145 PRINT AT R,14;" "
 147 NEXT R
 148 PAUSE 300
 150 CLS
 460 PRINT AT 6,0;"▓
▓"
 465 PRINT AT 7,0;"▓ CANONS (SYM
BOLE:UN NUMERO)  ▓"
 470 PRINT AT 8,0;"▓ (NOMBRE MAX
IMUM : 5)        ▓"
 475 PRINT AT 9,0;"▓
▓"
 480 PRINT AT 10,0;"▓ BOUCLIERS
SYMBOLE :----    ▓"
 482 PRINT AT 11,0;"▓ (NOMBRE MA
XIMUM : 3)       ▓"
 484 PRINT AT 12,0;"▓
▓"
 510 PRINT AT 21,2;"NB DE BOUCLI
ERS:"
 520 INPUT NB
 525 IF NB>3 OR NB<0 THEN GOTO 5
10
 530 CLS
 830 PRINT AT 4,0;"ENTREZ LA POS
ITION DE VOTRE BASE"
 840 PRINT AT 6,0;"ELLE SE TROUV
ERA SUR LA LIGNE 20"
 845 PRINT AT 8,0;"ATTENTION,X N
E DOIT PAS ETRE > A"
 846 PRINT AT 9,0;"14 NI INFERIE
UR A 0..."
 850 INPUT PB
 855 IF PB>14 OR PB<0 THEN GOTO
820
 942 LET PBZ=INT (RND*17)+13
 943 CLS
 945 PRINT AT 20,PB;"▓"
 946 PRINT AT 0,PBZ;"O"
 948 PRINT AT 1,PBZ-2;"-----"
 950 FOR R=0 TO 30
 955 PRINT AT INT (RND*3)+9,R;"▒
▒"
 960 PRINT AT INT (RND*3)+10,R;"
▒"
 965 PRINT AT INT (RND*3)+8,R;"▒
▒"
 966 NEXT R
 990 FOR M=1 TO INT (RND*7)+1
 993 PRINT AT INT (RND*11)+1,INT
(RND*31);"▓"
 995 PRINT AT INT (RND*11)+9,INT
(RND*31);"▓"
 998 NEXT M
 999 PRINT AT 21,0;"ACCEPTEZ-VOU
S LE TERRAIN DE JEU?"
1000 IF INKEY$="" THEN GOTO 1000
1001 IF INKEY$="O" THEN GOTO 100
3
1002 IF INKEY$<>"O" THEN GOTO 94
3
1004 PRINT AT 21,0;"POS. DES CAN
ONS (Y 13,21;X 0,31)"
1005 DIM Z(5)
1010 DIM W(5)
1020 FOR C=1 TO 5
1025 INPUT Z(C)
1026 IF Z(C)<13 OR Z(C)>20 THEN
GOTO 1025
1030 INPUT W(C)
1035 IF W(C)>31 OR W(C)<0 THEN G
OTO 1030
1040 PRINT AT Z(C),W(C);CHR$ (C+
165)
1050 NEXT C
1060 PRINT AT 21,0;"CORRECTION A
EFFECTUER ?    "
1065 IF INKEY$="" THEN GOTO 1065
1070 IF INKEY$="O" THEN GOTO 110
0
1080 GOTO 1190
1100 PRINT AT 21,0;"NUMERO DU CA
NON:      ?        "
1110 INPUT U
1111 PRINT AT Z(U),W(U);" "
1120 LET U1=U
1130 PRINT AT 21,0;"NOUVELLES PO
SITIONS (Y,X)      "
1140 INPUT Z(U1)
1145 IF Z(U1)<12 OR Z(U1)>20 THE
N GOTO 1140
1150 INPUT W(U1)
1155 IF W(U1)>31 OR W(U1)<0 THEN
 GOTO 1150
1160 PRINT AT Z(U1),W(U1);CHR$ (
U+165)
1170 GOTO 1060
1190 IF NB=0 THEN GOTO 1410
1200 PRINT AT 21,0;"ENTREZ LES P
OS.  DES BOUCLIERS  "
1210 DIM N(NB)
1220 DIM M(NB)
1230 FOR B=1 TO NB
1240 INPUT N(B)
1245 IF N(B)<12 OR N(B)>20 THEN
GOTO 1240
1250 INPUT M(B)
1255 IF M(B)>28 OR M(B)<0 THEN G
OTO 1250
1260 PRINT AT N(B),M(B);"--";B;"
-"
1265 NEXT B
1270 PRINT AT 21,0;"CORRECTIONS
A EFFECTUER ?      "
1280 IF INKEY$="" THEN GOTO 1280
1290 IF INKEY$="O" THEN GOTO 130
0
1295 GOTO 1400
1300 PRINT AT 21,0;"NUMERO DU BO
UCLIER:  ?         "
1310 INPUT V
1320 PRINT AT N(V),M(V);"    "
1330 PRINT AT 21,0;"NOUVELLES PO
SITIONS (Y,X)      "
1340 LET V1=V
1350 INPUT N(V1)
1355 IF N(V1)<12 OR N(V1)>20 THE
N GOTO 1350
1360 INPUT M(V1)
1365 IF M(V1)>28 OR M(V1)<0 THEN
 GOTO 1360
1370 PRINT AT N(V1),M(V1);"--";V
;"-"
1380 GOTO 1270
1400 PRINT AT 21,0;"JE PLACE : M
ES DEFENSES ...   "
1410 LET PP=PEEK 16396+PEEK 1639
7*256+1
1415 PRINT AT 6,0;"▓";AT 6,31;"▓
"
1425 DIM E(3)
1430 DIM F(3)
1435 LET E(1)=7
1440 LET F(1)=5
1445 LET E(2)=7
1450 LET F(2)=15
1455 LET E(3)=7
1460 LET F(3)=25
1470 PRINT AT E(1),F(1);"C";AT E
(2),F(2);"C";AT E(3),F(3);"C"
1500 IF NB=0 THEN GOTO 1580
1510 DIM G(NB)
1520 DIM H(NB)
1525 FOR J=1 TO NB
1530 LET G(J)=INT (RND*7)+1
1540 LET H(J)=INT (RND*28)
1545 IF PEEK (PP+(G(J)*33)+H(J))
=CODE "▓" THEN GOTO 1530
1550 IF PEEK (PP+(G(J)*33)+H(J))
=CODE "C" THEN GOTO 1530
1560 PRINT AT G(J),H(J);"----"
1570 NEXT J
1580 PRINT AT 21,0;"LES CHARS SE
 METTENT EN PLACE"
1582 PAUSE 120
```

```
1585 DIM Y(5)
1590 DIM X(5)
1600 LET Y(1)=20
1610 LET X(1)=15
1620 LET X(2)=17
1630 LET Y(2)=20
1640 LET X(3)=19
1650 LET Y(3)=20
1660 LET X(4)=21
1670 LET Y(4)=20
1680 LET X(5)=23
1690 LET Y(5)=20
1700 PRINT AT Y(1),X(1);"1";AT Y(2),X(2);"2";AT Y(3),X(3);"3";AT Y(4),X(4);"4";AT Y(5),X(5);"5"
1705 DIM D(6)
1706 DIM O(6)
1710 LET D(1)=2
1720 LET O(1)=0
1730 LET D(2)=4
1740 LET O(2)=0
1750 LET D(3)=6
1760 LET O(3)=0
1770 LET D(4)=8
1780 LET O(4)=0
1790 LET D(5)=10
1800 LET O(5)=0
1805 LET D(6)=12
1807 LET O(6)=0
1810 PRINT AT O(1),D(1);"▒";AT O(2),D(2);"▒";AT O(3),D(3);"▒";AT O(4),D(4);"▒";AT O(5),D(5);"▒";AT O(6),D(6);"▒"
1820 PRINT AT 21,0;"VOUS ENGAGEZ LE COMBAT..."
1830 FOR A=1 TO 40
1840 NEXT A
1852 IF PEEK (PP+660+PB)<>CODE "▒" THEN GOTO 9993
1855 PRINT AT 21,0;"U:JOUEZ C=CHAR,D=CANON,P=PASSE ?"
1860 IF INKEY$="" THEN GOTO 1860
1870 IF INKEY$="C" THEN GOTO 1900
1880 IF INKEY$="D" THEN GOTO 8000
1900 IF INKEY$="P" THEN GOTO 9000
1905 IF PEEK (PP+(Y(1)*33)+X(1))<>CODE "1" AND PEEK (PP+(Y(2)*33)+X(2))<>CODE "2" AND PEEK (PP+(Y(3)*33)+X(3))<>CODE "3" AND PEEK (PP+(Y(4)*33)+X(4))<>CODE "4" AND PEEK (PP+(Y(5)*33)+X(5))<>CODE "5" THEN GOTO 7450
1910 PRINT AT 21,0;"NUMERO DU CHAR JOUE ?"
1920 INPUT NU
1930 IF NU>5 OR NU<=0 THEN GOTO 1910
2020 IF PEEK (PP+(Y(NU)*33)+X(NU))<>CODE CHR$ (NU+28) THEN GOTO 7000
2025 PRINT AT 21,0;"DEPLACEMENT (K) OU TIR (F) ?"
2026 IF INKEY$="" THEN GOTO 2026
2030 IF INKEY$="F" THEN GOTO 2250
2035 IF INKEY$="K" THEN GOTO 2040
2037 GOTO 2020
2040 PRINT AT 21,0;"DIRECTION DE DEPLACEMENT ?"
2050 INPUT DA
2055 LET NA=INT (RND*5)+1
2057 PRINT AT Y(NU),X(NU);" "
2060 FOR A=1 TO NA
2062 LET VAR=INT (RND*5)
2065 LET YA2=((SIN (-DA*((2*PI)/360)))*A)
2066 LET XA2=((COS (DA*((2*PI)/360)))*A)
2067 IF PEEK ((PP+(YA2+Y(NU))*33+(XA2+X(NU))))=CODE "▒" THEN GOTO 7000
2068 IF PEEK ((PP+(YA2+Y(NU))*33+(XA2+X(NU))))=CODE "▒" THEN GOTO 7000
2069 IF PEEK ((PP+(YA2+Y(NU))*33+(XA2+X(NU))))=CODE "-" THEN GOTO 8990
2070 IF VAR=0 OR VAR=4 THEN GOTO 5000
2072 IF YA2+Y(NU)>20 OR YA2+Y(NU)<0 THEN GOTO 2086
2077 PRINT AT INT (YA2+Y(NU)),INT (XA2+X(NU));CHR$ (NU+28)
2080 PRINT AT INT (YA2+Y(NU)),INT (XA2+X(NU));" "
2086 NEXT A
2090 LET Y(NU)=INT (Y(NU)+YA2)
2092 LET X(NU)=INT (X(NU)+XA2)
2098 PRINT AT Y(NU),X(NU);CHR$ (NU+28)
2099 GOTO 9000
2250 PRINT AT 21,0;"DIRECTION DU TIR (EN DEGRES) ?"
2260 INPUT FA
2270 FOR A=1 TO 5
2275 LET FXA=INT ((COS (FA*((2*PI)/360)))*A)
2280 LET FYA=INT ((SIN (-FA*((2*PI)/360)))*A)
2300 IF (FYA+Y(NU))>20 OR (FYA+Y(NU))<0 THEN GOTO 2360
2350 PRINT AT (FYA+Y(NU)),(FXA+X(NU));"■"
2357 PRINT AT (FYA+Y(NU)),(FXA+X(NU));" "
2360 NEXT A
2370 GOTO 9000
5000 PRINT AT 6,0;"▒";AT 6,31;"▒"
5005 PRINT AT 6,1;"+++++++++++++++++++++++++++++++"
5010 PRINT AT 6,1;"                              "
5015 IF Y(NU)=6 THEN GOTO 7000
5020 GOTO 2077
5999 IF PEEK (PP+(O(L)*33)+D(L))<>CODE CHR$ (L+156) THEN GOTO 9026
6000 IF PEEK (PP+(Y(P)*33)+X(P))<>CODE CHR$ (P+28) THEN GOTO 9026
6001 LET I=O(L)
6002 LET Q=D(L)
6008 FOR S=1 TO 5
6010 LET D1=(SGN (Y(P)-I))
6020 LET D2=(SGN (X(P)-Q))
6025 LET I=I+D1
6026 LET Q=Q+D2
6030 PRINT AT I,Q;"■"
6035 PRINT AT I,Q;" "
6040 NEXT S
6042 GOTO 1850
6500 IF PEEK (PP+(O(L)*33)+D(L))<>CODE CHR$ (L+156) THEN GOTO 9030
6510 IF PEEK (PP+(Z(P)*33)+W(P))<>CODE CHR$ (P+165) THEN GOTO 9030
6520 LET I1=O(L)
6530 LET Q1=D(L)
6540 FOR S=1 TO 5
6545 LET D3=(SGN (Z(P)-I1))
6550 LET D4=(SGN (W(P)-Q1))
6555 LET I1=I1+D3
6560 LET Q1=Q1+D4
6565 PRINT AT I1,Q1;"■"
6570 PRINT AT I1,Q1;" "
6580 NEXT S
6585 GOTO 1850
6600 IF PEEK (PP+(O(L)*33)+D(L))<>CODE CHR$ (L+156) THEN GOTO 9025
6610 LET I3=O(L)
6620 LET Q3=D(L)
6630 FOR S=1 TO 5
6660 LET D5=(SGN (20-I3))
6670 LET D6=(SGN (PB-Q3))
6680 LET I3=I3+D5
6685 LET Q3=Q3+D6
6690 PRINT AT I3,Q3;"■"
6695 PRINT AT I3,Q3;" "
6700 NEXT S
6710 GOTO 1850
7000 PRINT AT 21,0;" LE CHAR ";NU;" A EXPLOSE..."
7010 POKE (PP+(Y(NU)*33)+X(NU)),0
7015 PAUSE 250
7020 GOTO 1850
7450 PRINT AT 21,0;"TOUS VOS CHARS SONT DETRUITS..."
7460 PAUSE 250
7470 GOTO 1850
8020 PRINT AT 21,0;"NUMERO DU CANON JOUE ?"
8030 INPUT CC
8032 IF CC>5 OR CC<=0 THEN GOTO 8020
8035 IF PEEK (PP+(Z(CC)*33)+W(CC))<>CODE CHR$ (CC+165) THEN GOTO 8150
8040 PRINT AT 21,0;"DIRECTION DE TIR ?"
8045 INPUT FC
8050 FOR C=1 TO 5
8060 LET FYC=INT ((SIN (-FC*((2*PI)/360)))*C)
8065 LET FXC=INT ((COS (FC*((2*PI)/360)))*C)
8070 IF (Z(CC)+FYC)>20 OR (Z(CC)+FYC)<0 THEN GOTO 8120
8100 PRINT AT Z(CC)+FYC,W(CC)+FXC;"■"
8110 PRINT AT Z(CC)+FYC,W(CC)+FXC;" "
8120 NEXT C
8130 GOTO 9000
8150 PRINT AT 21,0;" LE CANON A ETE DETRUIT..."
8160 PAUSE 250
8170 GOTO 1850
8990 PRINT AT 21,0;"BOUCLIER EXPLOSIF ... BRAMMMMM"
8992 PAUSE 250
9000 IF PEEK (PP+PBZ)<>CODE "O" THEN GOTO 9800
9003 PRINT AT 21,0;"****** JE RIPOSTE ******"
9010 PRINT AT 0,PBZ;"O"
9015 LET HA=INT (RND*6)+1
9018 IF PEEK (PP+(O(HA)*33)+D(HA))<>CODE CHR$ (HA+156) THEN GOTO 9000
9021 IF Y(1)<=12 AND Y(1)>=2 OR Y(2)<=12 AND Y(2)>=2 OR Y(3)<=12 AND Y(3)>=2 OR Y(4)<=12 AND Y(4
```

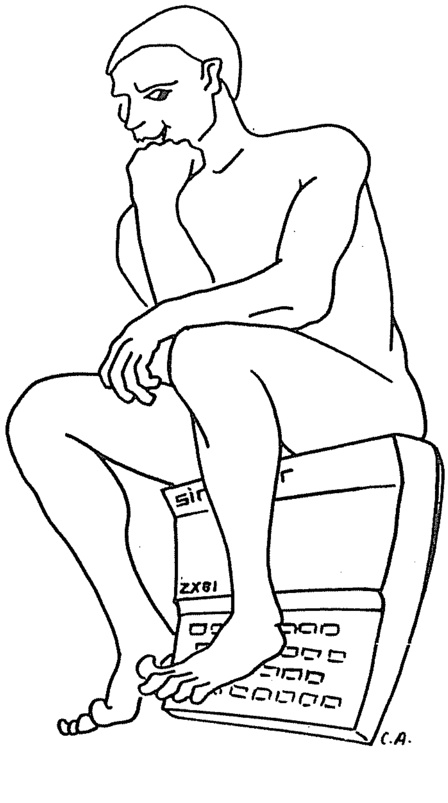
```
)>=2 OR Y(5)<=12 AND Y(5)>=2 THE
N GOSUB 9200
9022 FOR L=1 TO 6
9023 FOR P=1 TO 5
9024 IF ABS (O(L)-20)<=5 AND ABS
 (D(L)-PB)<=5 THEN GOTO 6600
9025 IF ABS (O(L)-Y(P))<=5 AND A
BS (D(L)-X(P))<=5 THEN GOTO 5999
9026 IF ABS (O(L)-Z(P))<=5 AND A
BS (D(L)-W(P))<=5 THEN GOTO 6500
9030 NEXT P
9036 NEXT L
9050 LET OO=INT (RND*5)+1
9060 PRINT AT O(HA),D(HA);" "
9065 FOR A=1 TO OO
9070 IF PEEK (PP+(Y(1)*33)+X(1))
<>CODE "1" AND PEEK (PP+(Y(2)*33
)+X(2))<>CODE "2" AND PEEK (PP+(
Y(3)*33)+X(3))<>CODE "3" AND PEE
K (PP+(Y(4)*33)+X(4))<>CODE "4"
AND PEEK (PP+(Y(5)*33)+X(5))<>CO
DE "5" THEN GOTO 9990
9080 LET DD1=(SGN (Y(INT (RND*5)
+1)-O(HA)))+(A-1)
9085 LET DD2=(SGN (X(INT (RND*5)
+1)-D(HA)))+(A-1)
9090 IF PEEK (PP+((O(HA)+DD1)*33
)+(D(HA)+DD2))=CODE "C" THEN GOS
UB 9600
9100 IF PEEK (PP+((O(HA)+DD1)*33
)+(D(HA)+DD2))=CODE "■" THEN GOT
O 9500
9105 IF (O(HA)+DD1)>20 THEN LET
O(HA)=0
9106 IF (O(HA)+DD1)<0 THEN LET O
(HA)=20
9107 IF (D(HA)+DD2)>31 THEN LET
D(HA)=0
9108 IF (D(HA)+DD2)<0 THEN LET D
(HA)=31
9110 PRINT AT O(HA)+DD1,D(HA)+DD
2;CHR$ (HA+156)
9120 PRINT AT O(HA)+DD1,D(HA)+DD
2;CHR$ 0
9125 NEXT A
9130 LET D(HA)=D(HA)+DD2
9140 LET O(HA)=O(HA)+DD1
9150 IF O(HA)=5 THEN GOTO 9033
9155 POKE (PP+(O(HA)*33)+D(HA)),
CODE CHR$ (HA+156)
9160 GOTO 1850
9170 POKE (PP+((O(HA)+1)*33)+D(H
A)),CODE CHR$ (HA+156)
9180 GOTO 1850
9200 FOR I=1 TO 3
9205 IF PEEK (PP+(E(I)*33)+F(I))
<>CODE "C" THEN GOTO 9250
9210 FOR V=1 TO INT (RND*5)+1
9211 IF PEEK (PP+(E(I)*33)+F(I)+
V)=CODE "■" OR PEEK (PP+(E(I)*33
)+F(I)+V)=CODE "■" OR PEEK (PP+(
E(I)*33)+F(I)+V)=CODE "■" OR PEE
K (PP+(E(I)*33)+F(I)+V)=CODE "■"
 OR PEEK (PP+(E(I)*33)+F(I)+V)=C
ODE "■" OR PEEK (PP+(E(I)*33)+F(
I)+V)=CODE "■" THEN GOTO 9221
9215 PRINT AT E(I),F(I)+V;"■"
9220 PRINT AT E(I),F(I)+V;" "
9221 IF PEEK (PP+(E(I)*33)+F(I)-
V)=CODE "■" OR PEEK (PP+(E(I)*33
)+F(I)-V)=CODE "■" OR PEEK (PP+(
E(I)*33)+F(I)-V)=CODE "■" OR PEE
K (PP+(E(I)*33)+F(I)-V)=CODE "■"
 OR PEEK (PP+(E(I)*33)+F(I)-V)=C
ODE "■" OR PEEK (PP+(E(I)*33)+F(
I)-V)=CODE "■" THEN GOTO 9227
9222 PRINT AT E(I),F(I)-V;"■"
9225 PRINT AT E(I),F(I)-V;" "
9227 IF PEEK (PP+((E(I)+V)*33)+F
(I))=CODE "■" OR PEEK (PP+((E(I)
+V)*33)+F(I))=CODE "■" OR PEEK (
PP+((E(I)+V)*33)+F(I))=CODE "■"
OR PEEK (PP+((E(I)+V)*33)+F(I))=
CODE "■" OR PEEK (PP+((E(I)+V)*3
3)+F(I))=CODE "■" OR PEEK (PP+((
E(I)+V)*33)+F(I))=CODE "■" THEN
GOTO 9233
9230 PRINT AT E(I)+V,F(I);"■"
9232 PRINT AT E(I)+V,F(I);" "
9233 IF PEEK (PP+((E(I)-V)*33)+F
(I))=CODE "■" OR PEEK (PP+((E(I)
-V)*33)+F(I))=CODE "■" OR PEEK (
PP+((E(I)-V)*33)+F(I))=CODE "■"
OR PEEK (PP+((E(I)-V)*33)+F(I))=
CODE "■" OR PEEK (PP+((E(I)-V)*3
3)+F(I))=CODE "■" OR PEEK (PP+((
E(I)-V)*33)+F(I))=CODE "■" THEN
GOTO 9245
9235 PRINT AT E(I)-V,F(I);"■"
9237 PRINT AT E(I)-V,F(I);" "
9245 NEXT V
9250 NEXT I
9260 RETURN
9540 PRINT AT 21,0;"UN DES CHARS
DU ZX A SAUTE...      "
9545 PAUSE 250
9550 GOTO 1850
9600 LET DD1=-DD1
9610 LET DD2=-DD2
9630 RETURN
9800 CLS
9810 PRINT AT 8,6;"LA BASE DU ZX
 A SAUTE..."
9820 PRINT AT 10,10;"BRAVO"
9830 GOTO 9994
9990 LET DD1=SGN (20-O(HA))+(A-1)
9991 LET DD2=SGN (PB-D(HA))+(A-1
)
9992 GOTO 9090
9993 CLS
9994 PRINT AT 10,0;"VOTRE BASE E
ST DETRUITE...      "
9995 PRINT AT 12,0;"POUR REJOUER
TAPEZ "R"           "
9996 IF INKEY$="" THEN GOTO 9994
9998 IF INKEY$="R" THEN RUN
9999 STOP
```

```
+2,M*2+5; PAPER 2;"□";AT N*2+2,M
*2+5;"■": NEXT W: FOR W=0 TO 60:
 PRINT AT N*2,M*2+3;"■";AT N*2,M
*2+3; PAPER 2;"□": NEXT W
3720 GO TO 235
3750 LET D=0: GO SUB 9550: IF D=
1 AND K=0 THEN GO TO 5122
3760 LET P(N,M)=8: LET P(N+1,M-1
)=0: FOR W=0 TO 60: PRINT AT N*2
+2,M*2+1; PAPER 2;"□";AT N*2+2,M
*2+1;"■": NEXT W: FOR W=0 TO 60:
 PRINT AT N*2,M*2+3;"■";AT N*2,M
*2+3; PAPER 2;"□": NEXT W
3780 GO TO 235
3800 FOR W=1 TO 60: PRINT AT X*2
,Y*2+3; PAPER 2;"□";AT X*2,Y*2+3
;"■": NEXT W: PRINT AT N*2,M*2+3
; PAPER 2;"□": FOR W=1 TO 60: PR
INT AT 4*N-2*X,4*M-2*Y+3;"■";AT
4*N-2*X,4*M-2*Y+3; PAPER 2;"□":
NEXT W
3805 GO SUB 9400
3807 GO SUB 9300
3810 GO TO 5000
3990 FOR N=0 TO 7: READ S: POKE
USR "A"+N,S: NEXT N
9992 DATA 231,195,129,0,0,129,19
5,231
9995 RETURN
```

F. Delaquaize

REVERSI-DAMES (SPECTRUM 16 K)

Un très beau reversi avec plein de couleurs et des pions qui flashent avant de se placer. Les règles du jeu, s'il en est besoin sont expliquées dans la liste.



```
  10 REM ©CLERGEOT Bertrand 1983
  12 GO SUB 9990
  15 PAPER 4: INK 0: CLS : BORDE
R 4
  20 GO SUB 6000
  30 PAPER 4: INK 0: CLS : BORDE
R 4: CLS
  95 PRINT
 100 FOR n=1 TO 5
 110 PRINT TAB 4; PAPER 7;"▄▄  ▄▄
▀▀  ▀▀  ▀▀  ▀▀  "
 120 PRINT TAB 4; PAPER 7;"██  ██
██  ██  ██  ██  "
 122 FOR N=1 TO 4
 160 PRINT TAB 4; PAPER 7;"▀▀▄▄
▀▀▄▄▀▀▄▄▀▀▄▄▀▀▄▄"
 161 PRINT TAB 4; PAPER 7;"  ██
██  ██  ██  ██  "
 162 PRINT TAB 4; PAPER 7;"▄▄▀▀
▄▄▀▀▄▄▀▀▄▄▀▀▄▄▀▀"
 163 PRINT TAB 4; PAPER 7;"██  ██
██  ██  ██  ██  ": NEXT N
 164 PRINT TAB 4; PAPER 7;"▀▀▄▄
▀▀▄▄▀▀▄▄▀▀▄▄▀▀▄▄"
 165 PRINT TAB 4; PAPER 7;"  ██
██  ██  ██  ██  "
 166 PRINT TAB 4; PAPER 7;"  ▀▀
▀▀  ▀▀  ▀▀  ▀▀  "
 167 PAPER 0
 168 DIM P(10,10)
 172 FOR N=0 TO 1: FOR M=1 TO 5:
 LET P(N*2+1,M*2-1)=1: LET P(10-
N*2-1,M*2-1)=8: PRINT AT N*4+2,M
*4+1; PAPER 5;"▣": PRINT AT 22-N
*4-2,M*4+3; PAPER 2;"▣": NEXT M
 180 NEXT N
 190 FOR N=0 TO 1: FOR M=1 TO 5:
 LET P(N*2+2,M*2)=1: LET P(10-N*
2,M*2)=8: PRINT AT N*4+4,M*4+3;
PAPER 5;"▣": PRINT AT 22-N*4-4,M
*4+1; PAPER 2;"▣": NEXT M
 200 NEXT N
 210 INK 0: PAPER 4: FOR N=1 TO
10: PRINT AT N*2,2;N: PRINT AT 0
,N*2+3;CHR$ (64+N): NEXT N
 212 GO SUB 9400
 220 INPUT "JE COMMENCE ? ";A$:
IF A$="O" THEN GO TO 5000
 235 GO SUB 9300
 238 BEEP .1,20
 240 INPUT "A1 enter B3 enter";A
$;B$: IF A$<>"FIN" AND B$<>"FIN"
 THEN LET Y1=CODE A$(1)-64: LET
Y2=CODE B$(1)-64
 242 IF A$="FIN" OR B$="FIN" THE
N GO TO 9000
 250 LET X1=VAL A$(2 TO ): LET X
2=VAL B$(2 TO )
 255 IF P(X1,Y1)<>1 THEN GO TO 2
40
 260 IF P(X2,Y2)<>0 OR X2<1 OR X
2>10 OR Y2<1 OR Y2>10 OR X1<1 OR
 X1>10 OR Y1<1 OR Y1>10 THEN GO
TO 240
 270 IF X2-X1=-1 OR ABS (X2-X1)>
2 OR ABS (Y1-Y2)>2 OR ABS (X1-X2
)<>ABS (Y1-Y2) THEN GO TO 240
 280 IF ABS (Y1-Y2)=2 AND P((X2-
X1)/2+X1,(Y2-Y1)/2+Y1)<>8 THEN G
O TO 240
 290 PRINT AT X1*2,Y1*2+3;"█";AT
 X2*2,Y2*2+3; PAPER 5;"▣"
 300 LET P(X1,Y1)=0: LET P(X2,Y2
)=1: IF ABS (Y1-Y2)=2 THEN LET P
((X2-X1)/2+X1,(Y2-Y1)/2+Y1)=1: P
RINT AT (X2-X1)+2*X1,(Y2-Y1)+2*Y
1+3; PAPER 5;"▣": GO TO 240
5000 FOR N=9 TO 2 STEP -1
5010 FOR M=9 TO 2 STEP -1
5020 IF P(N,M)=1 AND P(N-1,M-1)=
8 AND P(N+1,M+1)=0 THEN LET P(N+
1,M+1)=8: LET P(N,M)=8: LET P(N-
1,M-1)=0: LET X=N-1: LET Y=M-1:
GO TO 9800
5030 IF P(N,M)=1 AND P(N+1,M+1)=
8 AND P(N-1,M-1)=0 THEN LET P(N-
1,M-1)=8: LET P(N,M)=8: LET P(N+
1,M+1)=0: LET X=N+1: LET Y=M+1:
GO TO 9800
5040 IF P(N,M)=1 AND P(N-1,M+1)=
8 AND P(N+1,M-1)=0 THEN LET P(N+
1,M-1)=8: LET P(N,M)=8: LET P(N-
1,M+1)=0: LET X=N-1: LET Y=M+1:
GO TO 9800
5050 IF P(N,M)=1 AND P(N+1,M-1)=
8 AND P(N-1,M+1)=0 THEN LET P(N-
1,M+1)=8: LET P(N,M)=8: LET P(N+
1,M-1)=0: LET X=N+1: LET Y=M-1:
GO TO 9800
5060 NEXT M: NEXT N
5070 LET K=0
5080 FOR N=1 TO 9
5090 FOR M=10 TO 1 STEP -1
5100 IF M=1 THEN GO TO 9600
5110 IF M=10 THEN GO TO 9650
5120 IF P(N,M)=0 AND P(N+1,M-1)=
8 THEN GO TO 9750
5122 IF P(N,M)=0 AND P(N+1,M+1)=
8 THEN GO TO 9700
5130 NEXT M
5140 NEXT N
5145 IF K=1 THEN GO TO 9000
5150 LET K=1
5160 GO TO 5080
6010 DRAW 0,5: DRAW 70,42: PLOT
0,0: DRAW 255,0: DRAW 0,5: DRAW
-255,0: PLOT 255,5: DRAW -70,42
6015 FOR n=1 TO 10: PLOT n*25.5,
5: DRAW 11.5*n+70-n*25.5,42: NEX
T n
6020 PLOT 20,17: DRAW 215,0: PLO
T 38,28: DRAW 177,0: PLOT 52,37:
 DRAW 151,0: PLOT 62,43: DRAW 13
3,0: PLOT 69,47: DRAW 117,0
6025 PRINT AT 6,6; INK 1;"◆◆◆◆◆
◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆"
6028 PRINT AT 13,6; INK 1;"◆◆◆◆◆
◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆"
6030 FOR N=7 TO 12: PRINT AT N,6
; INK 1;"◆                  ◆":
NEXT N
6040 INK 2: PLOT 42,133: DRAW 17
2,0: DRAW 0,-75: DRAW -172,0: DR
AW 0,75
6045 PLOT 45,130: DRAW 166,0: DR
AW 0,-69: DRAW -166,0: DRAW 0,69
6050 PRINT AT 8,8; INK 0;"CLERGE
OT B. 1983"
6055 PRINT AT 10,10; INK 0;"Reve
rsi-Dame"
6070 IF INKEY$="" THEN GO TO 607
0
6072 BORDER 0: INK 1: PAPER 0: C
LS : OVER 1
7040 OVER 0: PLOT 40,50: DRAW 17
5,0: DRAW 0,95: DRAW -175,0: DRA
W 0,-95
7042 OVER 0: PLOT 37,47: DRAW 18
1,0: DRAW 0,101: DRAW -181,0: DR
AW 0,-101
7050 INK 5
7060 PRINT AT 6,8;"REGLE DU JEU"
7071 PRINT AT 8,6;"le deplacemen
t";AT 9,6;"est celui des dames";
AT 10,6;"mais une piece prise";A
T 11,6;"devient alliee.";AT 12,6
7080 IF INKEY$<>"" THEN GO TO 70
80
7090 IF INKEY$="" THEN GO TO 709
0
7100 PRINT AT 6,6;"Vous devez ca
pturer";AT 7,6;"les pieces rouge
s";AT 8,6;"le jeu fini lorsque";
AT 9,6;"la situation est     ";AT
 10,6;"bloquee: gagne celui";AT
11,6;"qui est majoritaire."
7110 IF INKEY$<>"" THEN GO TO 71
10
7120 IF INKEY$="" THEN GO TO 712
0
7130 PRINT AT 6,6;"lorsque vous
avez   ";AT 7,6;"pris, vous rejou
ez";AT 8,6;"vous n,etes pas     "
;AT 9,6;"obliger de prendre";AT
10,6;"                    ";AT 1
1,6;"    BONNE CHANCE    "
7200 IF INKEY$<>"" THEN GO TO 72
00
7210 IF INKEY$="" THEN GO TO 721
0
7230 RETURN
9000 GO SUB 9300: IF C<20 THEN P
RINT AT 0,9; FLASH 1; PAPER 6;"J
'AI GAGNE"
9020 IF C>20 THEN PRINT AT 0,5;
FLASH 1; PAPER 6;" VOUS AVEZ GA
GNE "
9030 IF C=20 THEN PRINT AT 0,4;
FLASH 1; PAPER 6;" LA PARTIE EST
 NULLE "
9040 STOP
9300 LET C=0: FOR N=1 TO 10: FOR
 M=1 TO 10: IF P(N,M)=1 THEN LET
 C=C+1
9310 NEXT M: NEXT N
9320 PRINT AT 1,27; BRIGHT 1;C;
PAPER 4; INK 5;" ◆";AT 3,27; BRI
GHT 1; INK 0;40-C; INK 2;" ◆"
9330 RETURN
9400 FOR N=7 TO 0 STEP -1: INK N
: PLOT 32,167: DRAW 167,0: DRAW
0,-167: DRAW -167,0: DRAW 0,167:
 NEXT N
9410 RETURN
9500 IF P(N-1,M-1)=1 OR (P(N+1,M
-1)=1 AND P(N-1,M+1)=0) OR (P(N-
1,M+1)=1 AND P(N+1,M-1)=0) THEN
LET D=1
9510 RETURN
9550 IF P(N-1,M+1)=1 OR (P(N+1,M
+1)=1 AND P(N-1,M-1)=0) OR (P(N-
1,M-1)=1 AND P(N+1,M+1)=0) THEN
LET D=1
9560 RETURN
9600 IF P(N+1,M+1)<>8 OR P(N,M)<
>0 THEN GO TO 5130
9610 GO TO 9710
9650 IF P(N+1,M-1)<>8 OR P(N,M)<
>0 THEN GO TO 5130
9660 GO TO 9760
9700 LET D=0: GO SUB 9500: IF D=
1 AND K=0 THEN GO TO 5130
9710 LET P(N,M)=8: LET P(N+1,M+1
)=0: FOR W=0 TO 60: PRINT AT N*2
```

B. Clergeot

LE JEU DE "QUI GAGNE" (ZX81 16K)

Un jeu de réflexion original et intéressant. L'ordinateur génère une grille de 10x10 remplie de chiffres de 1 à 9. Les règles du jeu sont expliquées dans le programme, il est essentiel que chaque joueur joue dans la colonne ou la ligne dans laquelle l'autre vient de jouer. L'ordinateur joue toujours le chiffre le plus grand.

```
  10 REM    QUI GAGNE
  20 REM
  30 REM
  40 REM
  50 REM    D,GUMY
  60 REM
  70 REM    (C)JUIN 1983
  80 REM
  90 REM    MACRO-LETTRES
 100 GOSUB 1000
 110 REM    REGLES DU JEU
 120 GOSUB 1500
 130 REM    INITIALISATION
 140 RAND
 150 DIM A(10,10)
 155 LET TOT=100
 160 LET CH=0
 170 LET X2=CH
 180 LET T1=CH
 190 LET T2=CH
 205 FAST
 210 PRINT AT 1,4;"ABCDEFGHIJ"
 220 PRINT AT 3,3;"            "
 230 PRINT AT 14,3;"            
"
 240 FOR I=1 TO 10
 250 PRINT AT I+3,0;I;AT I+3,3;"
 ";TAB 14;" "
 260 NEXT I
 270 REM    CHIFFRES DANS LA
 280 REM         GRILLE
 290 FOR L=1 TO 10
 300 FOR C=1 TO 10
 310 LET A(L,C)=INT (9*RND)+1
 315 PRINT AT L+3,C+3;A(L,C)
 320 NEXT C
 325 NEXT L
 330 SLOW
 335 PRINT AT 9,17;" TOTAL :"
 340 REM    JOUEUR
 350 PRINT AT 16,0;"COMMENCEZ-VO
US EN PREMIER (O/N)",,"?"
 360 IF INKEY$="" THEN GOTO 360
 365 IF INKEY$="O" THEN GOTO 390
 370 IF INKEY$="N" THEN GOTO 630
 380 GOTO 360
 390 IF NOT CH THEN GOTO 410
 400 PRINT AT 4,16;"JOUEZ SOIT";
AT 6,16;"LIGNE ";TAB 24;CH;AT 7,
16;"COLONNE ";TAB 24;CHR$ (CV+37
)
 410 PRINT AT 16,0;"    COORDONN
EE HORIZONTALE          ","?"
 420 INPUT AA
 430 IF AA<1 OR AA>10 THEN GOTO
410
 440 PRINT AT 16,15;"VERTICALE
"
 450 INPUT B$
 460 IF LEN B$>1 THEN GOTO 440
 470 LET BB=CODE B$-37
 480 IF BB<1 OR BB>10 THEN GOTO
440
 490 IF NOT CH THEN GOTO 510
 500 IF AA<>CH AND BB<>CV THEN G
OTO 410
 510 PRINT AT 4,16;"             "
 520 PRINT AT 6,16;"             "
 530 PRINT AT 7,16;"             "
 540 PRINT AT 16,0;"
                  ",""
 550 LET T1=T1+A(AA,BB)
 560 PRINT AT 12,17;"JOUEUR";TAB
25;T1
 570 LET A(AA,BB)=0
 580 LET CH=AA
 590 LET CV=BB
 600 PRINT AT CH+3,CV+3;"▒"
 610 LET TOT=TOT-1
 620 IF NOT TOT THEN GOTO 2500
 630 REM    ZX-81
 640 REM    GRILLE
 650 IF NOT CH THEN GOSUB 2000
 660 REM    CALCUL CC/DD
 670 GOSUB 2100
 720 LET T2=T2+A(CC,DD)
 730 PRINT AT 14,17;"ZX-81";TAB
25;T2
 740 LET A(CC,DD)=0
 750 LET CH=CC
 760 LET CV=DD
 770 PRINT AT CH+3,CV+3;"▒"
 780 LET TOT=TOT-1
 790 IF NOT TOT THEN GOTO 2500
 800 GOTO 400
1000 REM    SOUS-PROGRAMME
1005 FAST
1010 LET XX=20
1020 LET YY=37
1030 LET A$="QUI"
1040 GOSUB 1300
1050 LET XX=12
1060 LET YY=28
1070 LET A$="GAGNE"
1080 GOSUB 1300
1090 LET XX=28
1100 LET YY=19
1110 LET A$="?"
1120 GOSUB 1300
1125 SLOW
1130 LET A$="CONCEPTION : DIDIER
 GUMY"
1140 LET Y=LEN A$
1150 FOR I=Y TO 1 STEP -1
1160 FOR J=1 TO 31-Y+I
1170 PRINT AT 19,J;A$(I)
1180 PRINT AT 19,J-1;" "
1190 NEXT J
1200 NEXT I
1210 PRINT AT 21,0;"POUR CONTINU
ER PRESSEZ SUR ""C"""
1215 PAUSE 4E4
1220 IF INKEY$="" THEN GOTO 1220
1230 IF INKEY$<>"C" THEN GOTO 12
20
1250 CLS
1260 RETURN
1300 REM    SOUS-PROGRAMME
1305 LET Z$="▒RND4(▒ ▒ "
1310 FOR A=1 TO LEN A$
1315 LET XC=CODE A$(A)
1320 GOSUB 1340
1325 LET XX=XX+8
1330 NEXT A
1335 RETURN
1340 FOR Y=YY TO YY-7 STEP -1
1345 LET X$=CHR$ (PEEK (7680+8*X
C+YY-Y))
1350 FOR X=1 TO 8
1355 IF X$<Z$(X) THEN GOTO 1370
1360 PLOT XX+X-1,Y
1365 LET X$=CHR$ (CODE X$-CODE Z
$(X))
1370 NEXT X
1375 NEXT Y
1380 RETURN
1500 REM    SOUS-PROGRAMME
1510 PRINT AT 0,10;"QUI  GAGNE"
1520 PRINT AT 1,9;"------------"
1530 PRINT "VOUS DEVEZ CHOISIR U
N CHIFFRE","DANS LA GRILLE;SI JE
 JOUE AVANT","VOUS,VOUS DEVEZ JO
UER SUR LA","MEME LIGNE OU SUR L
A MEME","COLONNE QUE MOI."
1535 PRINT "LE SCORE LE PLUS HAU
T GAGNE","EVIDEMMENT",,"BONNE CH
ANCE."
1540 PRINT AT 21,0;"POUR CONTINU
ER PRESSEZ SUR ""C"""
1545 PAUSE 4E4
1550 IF INKEY$="" THEN GOTO 1550
1560 IF INKEY$<>"C" THEN GOTO 15
50
1570 CLS
1580 RETURN
2000 REM    SOUS-PROGRAMME
2010 LET CH=INT (RND*10)+1
2020 LET CV=INT (RND*10)+1
2030 RETURN
2100 REM    SOUS-PROGRAMME
2110 FOR N=9 TO 0 STEP -1
2120 FOR X=1 TO 10
2130 IF A(CH,X)=N THEN GOTO 2170
2140 IF A(X,CV)=N THEN GOTO 2200
2150 NEXT X
2160 NEXT N
2170 LET CC=CH
2180 LET DD=X
2190 RETURN
2200 LET CC=X
2210 LET DD=CV
2220 RETURN
2500 REM    RESULTAT
2510 PRINT AT 16,0;"    LA PARTI
E EST TERMINEE"
2520 IF T1>T2 THEN PRINT AT 18,0
;"      VOUS AVEZ GAGNE"
2530 IF T1<T2 THEN PRINT AT 18,0
;"      VOUS AVEZ PERDU"
2540 IF T1=T2 THEN PRINT AT 18,0
;"      PARTIE NULLE"
2550 PRINT AT 20,0;"    UNE NOUV
ELLE PARTIE (O/N)",,"?"
2560 IF INKEY$="" THEN GOTO 2560
2570 IF INKEY$="O" THEN GOTO 260
0
2580 IF INKEY$="N" THEN GOTO 265
0
2590 GOTO 2560
2600 CLEAR
2610 CLS
2620 GOTO 130
2650 CLS
2660 GOTO 9990
2700 REM    SAUVEGARDE
2710 SAVE "QUI GAGNE"
2720 GOSUB 1000
2730 PRINT AT 16,0;"VOULEZ-VOUS
LES REGLES DU JEU ",TAB 8;"(O/N)
?"
2740 IF INKEY$="" THEN GOTO 2740
2750 IF INKEY$="O" THEN GOTO 278
0
2760 IF INKEY$="N" THEN GOTO 280
0
2770 GOTO 2740
2780 CLS
2790 GOSUB 1500
2800 CLS
2810 GOTO 130
9990 REM    FIN
9995 PRINT AT 20,10;"MERCI"
9999 STOP
```

Didier Gumy

BOWLING (ZX81 16K)

Une belle réalisation en langage machine - liste garantie revérifiée 3 fois ! - Ce jeu utilise la manette de jeu simplifiée dont la construction a été proposée par ORDI-5 ; mais le jeu fonctionne aussi si l'on utilise les touches du clavier (toutes les touches sont efficaces, même BREAK). Attention la routine est rapide.




```
   5 REM
  10 POKE 16418,0
  20 FOR A=0 TO 41
  30 PRINT "BOWLING ";
  40 PRINT "BOWLING ";
  50 NEXT A
  60 PRINT ,,"        APPUYEZ SUR U
NE TOUCHE"
  70 IF INKEY$="" THEN GOTO 70
  80 LET A$=" BOWLI
NG"
  90 CLS
  92 POKE 16418,2
  95 LET L=USR 16514
 100 PRINT A$
 110 PRINT AT 10,3;"NOMBRE DE JO
UEURS ? (1 A 9)"
 120 LET I$=INKEY$
 130 IF I$="" THEN GOTO 020
 135 IF CODE I$<29 OR CODE I$>37
 THEN GOTO 120
 136 CLS
 137 PRINT A$
 140 LET N=VAL I$
 145 DIM S(N)
 150 DIM B$(N,20)
 160 FOR A=1 TO N
 170 PRINT AT 6+A,0;"NOM DU ";A;
".JOUEUR:";
 180 INPUT B$(A)
 190 PRINT B$(A)
 200 NEXT A
 210 CLS
 220 PRINT A$
 230 PRINT ,,,,"NOMBRE DE PARTIE
S PAR JOUEUR:";
 240 INPUT P
 260 CLS
 270 PRINT A$
 280 PRINT ,,,,"CHAQUE JOUEUR DI
SPOSE DE 8       BOULES PAR PARTI
E."
 290 PRINT ,,,,"APPUYEZ SUR UNE
TOUCHE"
 295 IF INKEY$="" THEN GOTO 295
 300 FOR A=1 TO N
 310 CLS
 320 PRINT A$
 330 PRINT ,,,,"A VOUS,";B$(A)
 340 PRINT
 350 PRINT "NIVEAU DE DIFFICULTE
?(1 A 255)"
 360 INPUT D
 370 IF D<1 OR D>255 THEN GOTO 3
50
 380 POKE 16417,256-D
 385 FOR B=1 TO P
 390 PRINT ,,,,"APPUYEZ SUR UNE
TOUCHE POUR      COMMENCER LE JEU
"
 395 IF INKEY$="" THEN GOTO 395
 400 POKE 16507,0
 410 CLS
 420 LET L=USR 16765
 430 LET S(A)=S(A)+PEEK 16507
 440 CLS
 445 IF B=P THEN GOTO 480
 450 PRINT AT 5,2;B;".PARTIE:";P
EEK 16507;" QUILLE";
 455 IF PEEK 16507>1 THEN PRINT
"S"
 460 PRINT TAB 2;"PRET POUR LA "
;B+1;".PARTIE ?"
 470 NEXT B
 480 PRINT AT 5,2;"BRAVO ";B$(A)
;TAB 2;"VOUS AVEZ TOTALISE ";S(A
);" QUILLE";
 481 IF S(A)>1 THEN PRINT "S"
 490 FOR Z=0 TO 100
 500 NEXT Z
 510 CLS
 520 NEXT A
 525 PRINT A$
 527 PRINT ,,,,
 530 FOR A=1 TO N
 545 PRINT B$(A);S(A);" QUILLE";
 547 IF S(A)>1 THEN PRINT "S"
 550 PRINT ""
 560 NEXT A
 570 PRINT ,,,,"VOULEZ-VOUS REJO
UER ?(O/N)"
 580 IF INKEY$="" THEN GOTO 580
 590 IF INKEY$="O" THEN GOTO 630
 600 PRINT ,,"AU REVOIR..."
 610 STOP
 620 SAVE "BOWLING"
 630 CLS
 640 RUN
```

```
16514,42,12,64,17,221,0,25,205
16522,211,64,17,137,0,25,205
16529,211,64,17,164,255,25,14
16536,15,205,4,65,17,208,7,27
16544,122,179,32,251,205,238,64
16551,35,13,32,238,62,189,205
16558,89,65,62,61,205,89,65,62
16566,189,205,89,65,62,61,205
16573,89,65,62,189,205,89,65,62
16581,61,205,89,65,62,189,205
16588,89,65,175,205,89,65,201
16595,229,54,128,35,54,128,17
16602,32,0,25,54,129,35,54,130
16610,6,3,25,54,128,35,54,128
16618,16,248,225,201,229,197,43
16625,17,27,0,14,7,6,6,54,0,35
16635,16,251,25,13,32,245,193
16642,225,201,229,54,131,35,6,2
16650,205,83,65,54,130,35,54,4
16658,17,28,0,25,54,133,35,6,5
16667,205,83,65,27,25,6,6,205
16675,83,65,54,5,25,6,6,205,83
16684,65,54,5,25,54,132,35,6,5
16693,205,83,65,54,1,25,54,2,35
16702,6,4,205,83,65,54,7,17,30
16711,0,25,54,3,35,54,3,35,54,1
16721,225,201,54,128,35,16,251
16728,201,42,12,64,35,14,3,6,0
16737,119,35,50,124,64,126,254
16744,118,58,124,64,32,1,35,16
16752,240,13,32,235,1,232,3,1
16760,120,177,32,251,201,175,50
16767,148,65,205,149,65,62,7,50
16775,119,65,205,205,65,58,119
16782,65,61,32,244,201,0,10,42
16790,12,64,17,9,0,62,136,205
16798,237,65,17,14,0,205,237,65
16806,17,243,255,62,128,205,237
16813,65,17,12,0,205,237,65,17
16821,247,255,25,17,5,7,66,229
16829,54,29,35,16,251,1,34,0
16837,225,9,21,40,3,21,24,238
16845,42,12,64,17,2,3,25,6,11
16854,54,52,58,33,64,79,205,250
16862,65,13,32,250,54,0,35,16
16870,239,17,245,255,25,24,231
16877,25,229,17,33,0,6,24,119
16885,25,16,252,225,201,229,197
16892,205,187,2,44,193,225,32,7
16900,219,254,254,255,40,1,201
16907,193,17,223,255,54,52,175
16914,1,184,11,11,120,177,32
16921,251,119,25,229,229,237,91
16928,12,64,33,254,255,167,237
16935,82,209,25,225,56,1,201
16942,126,254,29,204,54,66,24
16949,214,58,123,64,60,50,123
16956,64,229,58,148,65,254,0,40
16964,61,254,1,40,62,254,2,40
16972,63,254,3,40,64,254,4,40
16980,65,254,5,40,66,254,6,40
16988,67,254,7,40,68,254,8,40
16996,69,254,9,40,70,254,10,40
17004,71,254,11,40,72,254,12,40
17012,73,254,13,40,74,254,14,40
17020,75,17,111,2,24,73,17,2,0
17029,24,68,17,5,0,24,63,17,200
17038,0,24,58,17,203,0,24,53,17
17047,142,1,24,48,17,145,1,24
17055,43,17,84,2,24,38,17,87,2
17064,24,33,17,26,0,24,28,17,29
17073,0,24,23,17,224,0,24,18,17
17082,227,0,24,13,17,166,1,24,8
17091,17,169,1,24,3,17,108,2,42
17100,12,64,25,58,148,65,60,50
17108,148,65,205,211,64,225,201
```

M. Falque

Imprimerie Les Impressions CEPI - 75010 Paris - Imprimé en France - Directeur de la publication Bernard Savonnet - Dépôt légal : décembre 83



L'INCROYABLE MICRO-ORDINATEUR COULEUR SECAM !

- *Microprocesseur Z 80 A*
- *Langage Microsoft Basic*
- *Affichage direct antenne télé SECAM*
- *Clavier 45 touches pleine écriture, + clef d'entrée, + graphismes, + bip sonore anti-erreurs...*
- *Texte + graphismes mixables 9 couleurs*
- *Edition et correction plein écran*
- *Son incorporé*
- *Toutes options : extension + 16 K + 64 K, interface imprimante, imprimante, stylo optique, manettes, jeux, modem, disquettes...*

VTECH

VIDEO TECHNOLOGIE FRANCE
19, rue Luisant 91310 Monthléry
Tél. (6) 901.93.40 - Télex : SIGMA 180114

BON DE COMMANDE

A retourner à : VIDEO TECHNOLOGIE - 19, rue Luisant - 91310 Monthléry - Tél. (6) 901.93.40 - Télex SIGMA 180114

Je désire recevoir:

☐ **Version A**

Micro-ordinateur couleur SECAM LASER 200 ......990 F TTC

Kit d'accessoires:

- Modulateur SECAM incorporé
- \+ Transfo 220 V 50 HZ
- \+ 3 interfaces : câble télé, câble vidéo, câble lecteur K7
- \+ Livre utilisateur Basic en français, 150 pages
- \+ Livrets techniques en français
- \+ Cassette
- \+ Garantie 1 an, pièces et main-d'œuvre

Le kit complet .... 290 F TTC

1.280 F TTC

Extensions - Périphériques - Interfaces

☐ Extension de mémoire 16 K RAM (soit 20 K disponibles) 540 F TTC
☐ Extension de mémoire 64 K RAM (soit 68 K disponibles) *(livraison fin octobre)* .... 990 F TTC
☐ Lecteur de cassette DR 10 .... 490 F TTC
☐ Interface d'imprimante « Centronics » .... 290 F TTC
☐ Imprimante 4 couleurs *(livraison fin septembre)* ... 2.360 F TTC
☐ Manettes de jeux (la paire) *(livraison fin septembre)* . 290 F TTC
☐ Stylo lumineux *(livraison fin octobre)* .... N.C.
☐ Interface disquette *(livraison fin octobre)* .... N.C.

TOTAL DE MA COMMANDE : ______ F TTC

Nom ____________________
Prénom ____________________
N° ________ Rue ____________________
Ville ____________________
Code Postal |_|_|_|_|_|

Je choisis de payer le total de ma commande :

☐ Au comptant, par CCP, chèque bancaire ou mandat, à l'ordre de VIDEO TECHNOLOGIE FRANCE.
☐ Contre-remboursement au transporteur, moyennant une taxe de 60 F.

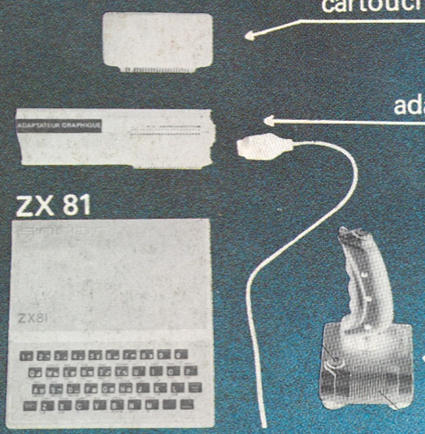
Signature

Au cas où je ne serais pas entièrement satisfait, je suis libre pendant un délai de 15 jours, de retourner à mes frais dans son emballage d'origine le matériel que j'aurai reçu et je serai intégralement remboursé des sommes que j'aurai versées.

ORDI 5

VTR Software

54, rue Ramey 75018 PARIS téléphone : 252.87.97





CARTOUCHES DISPONIBLES ACTUELLEMENT

nos cartouches de jeux travaillent en haute résolution grâce à l'adaptateur graphique sur un ZX 81 de base IK.

elles sont également compatibles avec la carte SAM (couleur) et la carte VTR (son).

le jeu est alors en haute résolution couleur et sonore !

DEMANDEZ NOTRE CATALOGUE
GÉNÉRAL SOFTWARE
CARTOUCHES ET CASSETTES
pour
ZX 81, ORIC, SPECTRUM, VIC 20
COMMODORE 64

VOUS CONNECTEZ LA CARTOUCHE et VOUS JOUEZ !



Une sélection des meilleurs logiciels...
Les noms les plus prestigieux

- CASES COMPUTER SIMULATIONS
- Mr. CHIP SOFTWARE
- HEWSON CONSULTANTS
- IMAGINE
- INTERCEPTOR MICRO'S
- J K GREY ENTERPRISES
- LLAMASOFT
- MARTECH GAMES
- MIKROGEN
- QUICK SILVA
- R and R SOFTWARE
- I J K SOFTWARE
- NEW GENERATION SOFTWARE

Les produits de VTR Software sont disponibles dans les points de vente VTR Informatique.

Magasin de vente : Même adresse.
Horaires : 10 h 30 - 13 h 30 et 15 h - 19 h.
Jours d'ouverture : du mardi au samedi inclus.
Métro : Jules Joffrin ou Marcadet Poissonniers

VTR Software
est un Département de Vidéo Telemat Report Sarl

Pour recevoir notre catalogue, remplissez le coupon ci-dessous et retournez-le accompagné de 5 F en timbres, à VTR Software, 54 rue Ramey, 75018 PARIS.

DEMANDE DE CATALOGUE

Nom : ______ Prénom : ______

Adresse : ______

Code Postal : ______